Filipe Manuel Pereira Duarte Rodrigues

Probabilistic models for learning from crowdsourced data

Doctoral thesis submitted to the Doctoral Program in Information Science and Technology,
supervised by Full Professor Francisco Camara Pereira and Professor Bernardete Ribeiro,
and presented to the Department of Informatics Engineering
of the Faculty of Sciences and Technology of the University of Coimbra.

September 2015

· U C ·

Universidade de Coimbra

Probabilistic models for learning from crowdsourced data

A thesis submitted to the University of Coimbra
in partial fulfillment of the requirements for the
Doctoral Program in Information Science and Technology

by

Filipe Manuel Pereira Duarte Rodrigues
fmpr@dei.uc.pt

Department of Informatics Engineering
Faculty of Sciences and Technology
University of Coimbra

September 2015

Financial support by
Fundação para a Ciência e a Tecnologia
Ref.: SFRH/BD/78396/2011
Probabilistic models for learning from crowdsourced data

Cover image: © Guilherme Nicholas
(http://www.flickr.com/photos/guinicholas/19768953491/in/faves-59695809@N06/)

****
****

Advisors

Prof. Francisco Câmara Pereira

Full Professor
Department of Transport
Technical University of Denmark

Prof. Bernardete Martins Ribeiro

Associate Professor with Aggregation
Department of Informatics Engineering
Faculty of Sciences and Technology of University of Coimbra

Dedicated to my parents

Abstract

This thesis leverages the general framework of probabilistic graphical models to develop probabilistic approaches for learning from crowdsourced data. This type of data is rapidly changing the way we approach many machine learning problems in different areas such as natural language processing, computer vision and music. By exploiting the wisdom of crowds, machine learning researchers and practitioners are able to develop approaches to perform complex tasks in a much more scalable manner. For instance, crowdsourcing platforms like Amazon mechanical turk provide users with an inexpensive and accessible resource for labeling large datasets efficiently. However, the different biases and levels of expertise that are commonly found among different annotators in these platforms deem the development of targeted approaches necessary.

With the issue of annotator heterogeneity in mind, we start by introducing a class of latent expertise models which are able to discern reliable annotators from random ones without access to the ground truth, while jointly learning a logistic regression classifier or a conditional random field. Then, a generalization of Gaussian process classifiers to multiple-annotator settings is developed, which makes it possible to learn non-linear decision boundaries between classes and to develop an active learning methodology that is able to increase the efficiency of crowdsourcing while reducing its cost. Lastly, since the majority of the tasks for which crowdsourced data is commonly used involves complex high-dimensional data such as images or text, two supervised topic models are also proposed, one for classification and another for regression problems. Using real crowdsourced data from Mechanical Turk, we empirically demonstrate the superiority of the aforementioned models over state-of-the-art approaches in many different tasks such as classifying posts, news stories, images and music, or even predicting the sentiment of a text, the number of stars of a review or the rating of movie.

But the concept of crowdsourcing is not limited to dedicated platforms such as Mechanical Turk. For example, if we consider the social aspects of the modern Web, we begin to perceive the true ubiquitous nature of crowdsourcing. This opened up an exciting new world of possibilities in artificial intelligence. For instance, from the perspective of intelligent transportation systems, the information shared online by crowds provides the context that allows us to better understand how people move in urban environments. In the second part of this thesis, we explore the use of data generated by crowds as additional inputs in order to improve machine learning models. Namely, the problem of understanding public transport demand in the presence of special events such as concerts, sports games or festivals, is considered. First, a probabilistic model is developed for explaining non-habitual overcrowding using crowd-generated information mined from the Web. Then, a Bayesian additive model with Gaussian process components is proposed. Using real data from Sin-

gapore's transport system and crowd-generated data regarding special events, this model is empirically shown to be able to outperform state-of-the-art approaches for predicting public transport demand. Furthermore, due to its additive formulation, the proposed model is able to breakdown an observed time-series of transport demand into a routine component corresponding to commuting and the contributions of individual special events.

Overall, the models proposed in this thesis for learning from crowdsourced data are of wide applicability and can be of great value to a broad range of research communities.

Keywords: probabilistic models, crowdsourcing, multiple annotators, transport demand, urban mobility, topic modeling, additive models, Bayesian inference

Resumo

A presente tese propõe um conjunto de modelos probabilísticos para aprendizagem a partir de dados gerados pela multidão (*crowd*). Este tipo de dados tem vindo rapidamente a alterar a forma como muitos problemas de aprendizagem máquina são abordados em diferentes áreas do domínio científico, tais como o processamento de linguagem natural, a visão computacional e a música. Através da sabedoria e conhecimento da *crowd*, foi possível na área de aprendizagem máquina o desenvolvimento de abordagens para realizar tarefas complexas de uma forma muito mais escalável. Por exemplo, as plataformas de *crowdsourcing* como o Amazon mechanical turk (AMT) colocam ao dispor dos seus utilizadores um recurso acessível e económico para etiquetar largos conjuntos de dados de forma eficiente. Contudo, os diferentes vieses e níveis de perícia individidual dos diversos anotadores que contribuem nestas plataformas tornam necessário o desenvolvimento de abordagens específicas e direcionadas para este tipo de dados multi-anotador.

Tendo em mente o problema da heterogeneidade dos anotadores, começamos por introduzir uma classe de modelos de conhecimento latente. Estes modelos são capazes de diferenciar anotadores confiáveis de anotadores cujas respostas são dadas de forma aleatória ou pouco premeditada, sem que para isso seja necessário ter acesso às respostas verdadeiras, ao mesmo tempo que é treinado um classificador de regressão logística ou um *conditional random field*. De seguida, são considerados modelos de crescente complexidade, desenvolvendo-se uma generalização dos classificadores baseados em processos Gaussianos para configurações multi-anotador. Estes modelos permitem aprender fronteiras de decisão não lineares entre classes, bem como o desenvolvimento de metodologias de aprendizagem activa, que são capazes de aumentar a eficiência do *crowdsourcing* e reduzir os custos associados. Por último, tendo em conta que a grande maioria das tarefas para as quais o *crowdsourcing* é usado envolvem dados complexos e de elevada dimensionalidade tais como texto ou imagens, são propostos dois modelos de tópicos supervisionados: um, para problemas de classificação e, outro, para regressão. A superioridade das modelos acima mencionados sobre as abordagens do estado da arte é empiricamente demonstrada usando dados reais recolhidos do AMT para diferentes tarefas como a classificação de posts, notícias, imagens e música, ou até mesmo na previsão do sentimento latente num texto e da atribuição do número de estrelas a um restaurante ou a um filme.

Contudo, o conceito de *crowdsourcing* não se limita a plataformas dedicadas como o AMT. Basta considerarmos os aspectos sociais da Web moderna, que rapidamente começamos a compreender a verdadeira natureza ubíqua do *crowdsourcing*. Essa componente social da Web deu origem a um mundo de possibilidades estimulantes na área de inteligência artificial em geral. Por exemplo, da perspectiva dos sistemas inteligentes de transportes, a informação partilhada online por multidões

fornece o contexto que nos dá a possibilidade de perceber melhor como as pessoas se movem em espaços urbanos. Na segunda parte desta tese, são usados dados gerados pela *crowd* como entradas adicionais de forma a melhorar modelos de aprendizagem máquina. Nomeadamente, é considerado o problema de compreender a procura em sistemas de transportes na presença de eventos, tais como concertos, eventos desportivos ou festivais. Inicialmente, é desenvolvido um modelo probabilístico para explicar sobrelotações anormais usando informação recolhida da Web. De seguida, é proposto um modelo Bayesiano aditivo cujas componentes são processos Gaussianos. Utilizando dados reais do sistema de transportes públicos de Singapura e dados gerados na Web sobre eventos, verificamos empiricamente a qualidade superior das previsões do modelo proposto em relação a abordagens do estado da arte. Além disso, devido à formulação aditiva do modelo proposto, verificamos que este é capaz de desagregar uma série temporal de procura de transportes numa componente de rotina (e.g. devido à mobilidade pendular) e nas componentes que correspondem às contribuições dos vários eventos individuais identificados.

No geral, os modelos propostos nesta tese para aprender com base em dados gerados pela *crowd* são de vasta aplicabilidade e de grande valor para um amplo espectro de comunidades científicas.

Palavras-chave: modelos probabilísticos, crowdsourcing, múltiplos anotadores, mobilidade urbana, modelos de tópicos, modelos aditivos, inferência Bayesiana

Acknowledgements

I wish to thank my advisors Francisco Pereira and Bernardete Ribeiro for inspiring and guiding me through this uncertain road, while giving me the freedom to satisfy my scientific curiosity. Francisco Pereira was the one who introduced me to the research world. Since then, he has been an exemplary teacher, mentor and friend. Bernardete Ribeiro joined us a little later but, since then, her guidance and mentorship have been precious and our discussions invaluable. I cannot thank them enough.

I would also like to thank my friends and family for their friendship and support. I especially thank my parents for their endless love and care. A huge part of what I am today, I owe to all of them.

I am also grateful to the MIT Intelligent Transportation Systems (ITS) lab and the Singapore MIT-Alliance for Research and Technology (SMART) centre for hosting me in several occasions. It was a pleasure to work and exchange ideas with everyone there.

Finally, the research that led to this thesis would not have been possible without the funding and support provided by the Fundação para a Ciência e Tecnologia under the scholarship SFRH/BD/78396/2011, by the research projects CROWDS (FCT - PTDC/EIA-EIA/115014/2009) and InfoCROWDS (FCT - PTDC/ECM-TRA/1898/2012), and by the Centre for Informatics and Systems of University of Coimbra (CISUC).

Notation

The notation used in this thesis is intended to be consistent and intuitive, while being as coherent as possible with the state of the art. In order to achieve that, symbols can sometimes change meaning during one or more chapters. However, such changes will be explicitly mentioned in the text, so that the meaning of a given symbol is always clear from the context.

Lowercase letters, such as x, represent variables. Vectors are denoted by bold letters such as $\mathbf{x}$, where the n^{th} element is referred as x_n. All vectors are assumed to be column vectors. Uppercase Roman letters, such as N, denote constants, and matrices are represented by bold uppercase letters such as $\mathbf{X}$. A superscript T denotes the transpose of a matrix or vector, so that $\mathbf{x}^{\mathrm{T}}$ will be a row vector.

Consider a discrete variable z that can take K possible values. It will be often convenient to represent z using a 1-of-K (or *one-hot*) coding scheme, in which z is a vector of length K such that if the value of the variable is j, then all elements z_k of z are zero except element z_j, which takes the value 1. Regardless of its coding scheme, a variable will always be denoted by a lowercase non-bold letter.

If there exist N values $\mathbf{x}_1, ..., \mathbf{x}_N$ of a D-dimensional vector $\mathbf{x} = (x_1, ..., x_D)^{\mathrm{T}}$, the observations can be combined into an $N \times D$ data matrix $\mathbf{X}$ in which the n^{th} row of $\mathbf{X}$ corresponds to the row vector $\mathbf{x}_n^{\mathrm{T}}$. This is convenient for performing operations over this matrix, thus allowing the representation of certain equations to be more compact. However, in some situations it will be necessary to refer to groups of matrices and vectors. A simple and intuitive way of doing so, is by using ranges in the subscript. Hence, if $\boldsymbol{\beta}_k$ is a vector, the collection of all $\{\boldsymbol{\beta}_k\}_{k=1}^{K}$ can simply be referred as $\boldsymbol{\beta}_{1:K}$. Similarly, the collection of matrices $\{\mathbf{M}_n\}_{n=1}^{N}$ can be denoted as $\mathbf{M}_{1:N}$. This notation provides a non-ambiguous and intuitive way of denoting collections of vectors and matrices without introducing new symbols and keeps the notation uncluttered.

The following tables summarize the notation used in this thesis.

General mathematical notation

Symbol	Meaning
$\triangleq$	Defined as
$\propto$	Proportional to, so $y = ax$ can be written as $y \propto x$
∇	Vector of first derivatives
$\exp(x)$	Exponential function, $\exp(x) = e^x$
$\mathbb{I}(x)$	Indicator function, $\mathbb{I}(x) = 1$ if x is true, otherwise $\mathbb{I}(x) = 0$
$\delta_{x,x'}$	Kronecker delta function, $\delta_{x,x'} = 1$ if $x = x'$ and $\delta_{x,x'} = 0$ otherwise
$\Gamma(x)$	Gamma function, $\Gamma(x) = \int_0^\infty u^{x-1} e^{-u} du$
$\Psi(x)$	Digamma function, $\Psi(x) = \frac{d}{dx} \log \Gamma(x)$

Linear algebra notation

Symbol	Meaning
$\text{tr}(\mathbf{X})$	Trace of matrix $\mathbf{X}$
$\det(\mathbf{X})$	Determinant of matrix $\mathbf{X}$
$\mathbf{X}^{-1}$	Inverse of matrix $\mathbf{X}$
$\mathbf{X}^{\text{T}}$	Transpose of matrix $\mathbf{X}$
$\mathbf{x}^{\text{T}}$	Transpose of vector $\mathbf{x}$
$\mathbf{I}_D$	Identity matrix of size $D \times D$
$\mathbf{1}_D$	Vector of ones with length D
$\mathbf{0}_D$	Vector of zeros with length D

Probability notation

Symbol	Meaning
$p(x)$	Probability density or mass function
$p(x\|y)$	Conditional probability density of x given y
$x \sim p$	x is distributed according to distribution p
$\mathbb{E}_q[x]$	Expected value of x (under the distribution q)
$\mathbb{V}_q[x]$	Variance of x (under the distribution q)
$\text{cov}[\mathbf{x}]$	Covariance of $\mathbf{x}$
$\mathbb{KL}(p\|\|q)$	Kullback-Leibler divergence, $\mathbb{KL}(p\|\|q) = \int p(x) \log \frac{p(x)}{q(x)}$
$\Phi(x)$	cumulative unit Gaussian, $\Phi(x) = (2\pi)^{-1/2} \int_{-\infty}^{x} \exp(-u^2/2) du$
$\text{Sigmoid}(x)$	Sigmoid (logistic) function, $\text{Sigmoid}(x) = 1/(1+e^{-x})$
$\text{Softmax}(\mathbf{x}, \boldsymbol{\eta})$	Softmax function, $\text{Softmax}(\mathbf{x}, \boldsymbol{\eta})_c = \frac{\exp(\boldsymbol{\eta}_c^{\text{T}}\mathbf{x})}{\sum_l \exp(\boldsymbol{\eta}_l^{\text{T}}\mathbf{x})}$, for $c \in \{1, \ldots, C\}$

Machine learning notation

Symbol	Meaning
$\mathbf{x}_n$	n^{th} instance
c_n	true class for the n^{th} instance
y_n^r	label of the r^{th} annotator for the n^{th} instance
α^r	sensitivity of the r^{th} annotator (Chapters 3 and 4)
β^r	specificity of the r^{th} annotator (Chapters 3 and 4)
$\boldsymbol{\eta}$	regression coefficients or weights
N	number of instances
R	number of annotators
C	number of classes
$\mathcal{D}$	dataset
$\boldsymbol{\Pi}^r$	reliability parameters of the r^{th} annotator
$\pi_{c,l}^r$	probability that the r^{th} annotator provides the label l given that the true class is c
z_n^r	latent reliability indicator variable (Chapter 3)
ϕ^r	reliability parameter for the r^{th} annotator (Chapter 3)
Z or $Z(\cdot)$	normalization constant
K	number of feature functions (Chapter 3); number of topics (Chapter 5)
T	length of the sequence (Chapter 3)

Symbol	Meaning
f_n	function value for the n^{th} instance, $f(\mathbf{x}_n)$
ϵ	observation noise
$m(\mathbf{x})$	mean function
$k(\mathbf{x}, \mathbf{x}')$	covariance function
$\mathbf{K}_N$	$N \times N$ covariance matrix
$\mathbf{x}_*$	test point
f_*	function value for the test instance, $f(\mathbf{x}_*)$
$\mathbf{k}_*$	vector with the covariance function evaluated between the test point $\mathbf{x}_*$ and all the points $\mathbf{x}$ in the dataset
k_{**}	covariance function evaluated between the test point $\mathbf{x}_*$ and itself
$\mathbf{V}_N$	$N \times N$ covariance matrix with observation noise included
D	dimensionality of the input space (Chapter 4); number of documents in the dataset (Chapter 5)
$\boldsymbol{\beta}_k$	distribution over words of the k^{th} topic
$\boldsymbol{\theta}^d$	topic proportions of the d^{th} document
z_n^d	topic assignment for the n^{th} word in the d^{th} document
w_n^d	n^{th} word in the d^{th} document
N_d	number of words in the d^{th} document (Chapter 5)
D_r	number of documents labeled by the r^{th} annotator (Chapter 5)
V	size of the word vocabulary (Chapter 5)
α	parameter of the Dirichlet prior over topic proportions (Chapters 5)
τ	parameter of the Dirichlet prior over the topics' distribution over words
ω	parameter of the Dirichlet prior over the reliabilities of the annotators
$\bar{\mathbf{z}}^d$	mean topic-assignment for the d^{th} document
c^d	true class for the d^{th} document
$y^{d,r}$	label of the r^{th} annotator for the d^{th} document
x^d	true target value for the d^{th} document
b^r	bias of the r^{th} annotator (Chapter 5)
p^r	precision of the r^{th} annotator (Chapter 5)
$\mathcal{L}$	evidence lower bound
h_n	n^{th} hotspot impact (Chapter 6)
a_n	n^{th} non-explainable component (Chapter 6)
b_n	n^{th} explainable component (Chapter 6)
e_n^i	contribution of the i^{th} event on the n^{th} observation (Chapter 6); i^{th} event associated on the n^{th} observation (Chapter 7)
β_a	variance associated with the non-explainable component a_n (Ch. 6)
β_e	variance associated with the event component e_n^i (Chapter 6)
$\mathbf{x}_n^r$	routine features associated with the n^{th} observation
$\mathbf{x}_n^{e_i}$	features of the i^{th} event associated with the n^{th} observation
E_n	number of events associated with the n^{th} observation
y_n^r	contribution of the routine components to the the n^{th} observation
$y_n^{e_i}$	contribution of the i^{th} event to the the n^{th} observation
β_r	variance associated with the routine component y_n^r (Chapter 7)

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Meaning
AMT	Amazon mechanical turk
API	application programming interface
ARD	automatic relevance determination
BART	Bayesian additive regression trees
CDF	cumulative distribution function
CRF	conditional random field
DiscLDA	discriminative latent Dirichlet allocation
DMR	Dirichlet-multimonial regression
EP	expectation propagation
EM	expectation-maximization
GP	Gaussian process
GPS	Global Positioning System
HMM	hidden Markov model
IRTM	inverse regression topic model
LDA	latent Dirichlet allocation
LTA	Land and Transport Authority
MAE	mean absolute error
MAP	maximum-a-posteriori
MCMC	Markov chain Monte Carlo
MedLDA	Maximum-entropy discrimination LDA
MLE	Maximum-likelihood estimation
MNIR	multinomial inverse regression
NER	named entity recognition
NFC	Near Field Communication
NLP	natural language processing
NP	noun phrase
POS	part-of-speech
RFID	Radio-frequency Identification
RMSE	root mean squared error
RRSE	root relative squared error
sLDA	supervised latent Dirichlet allocation
SVM	support vector machine
VBEM	variational Bayesian EM
w.r.t.	with respect to

Chapter 1

Introduction

1.1 Motivation

The origins of the field of machine learning can be traced back to the middle of the 20th century. However, it was not until the early 1990s that it started to have a significant and widespread practical impact, with the development of many successful applications in various research domains ranging from autonomous vehicles to speech recognition. This success can be justified by several factors such as the development of improved algorithms or the growing availability of inexpensive computers with an ever-increasing processing power. But perhaps the most important driving factor of this success was the exponential increase of data being gathered and stored. However, while this provides researchers with an unprecedented potential for solving complex problems, the growing sizes of modern datasets also pose many interesting challenges to the machine learning community. It is in this data-driven world that crowdsourcing plays a vital role.

Crowdsourcing (Howe, 2008) is the act of someone taking a task once performed by a single individual and outsourcing it to an undefined and generally large network of people. By relying on information produced by large crowds, crowdsourcing is rapidly redefining the way we approach many machine learning problems and the way that datasets are built. Through crowdsourcing, machine learning researchers and practitioners are able to exploit the wisdom of crowds to teach machines how to perform complex tasks in a much more scalable manner.

Let us consider the subclass of machine learning tasks corresponding to supervised learning problems. In supervised learning, the goal is to learn a mapping from inputs $\mathbf{x}$ to outputs y, given a labeled set of input-output pairs. Given a supervised learning problem, there are two ways in which crowdsourced data can be used to build predictive models: by using the labels provided by multiple annotators as a replacement for the true outputs y when these are hard or expensive to obtain, or by using information provided by crowds as additional input features that can help the model to understand the mapping from inputs $\mathbf{x}$ to outputs y. In this thesis, we develop probabilistic models for learning from crowdsourced data in both of these settings. Each of them provides its own set of challenges. However, as we shall see next, they are both of great practical importance.

A very popular way of applying crowdsourcing to machine learning problems is through the use of multiple annotators and crowds to label large datasets. With the development and proliferation of crowdsourcing platforms such as Amazon me-

chanical turk (AMT)[1] and CrowdFlower[2], it is becoming increasingly easier to obtain labeled data for a wide range of tasks from different areas such as computer vision, natural language processing, speech recognition, music, etc. The attractiveness of these platforms comes not only from their low cost and accessibility, but also from the surprisingly good quality of the labels obtained, which has been shown to compete with that of labels provided by "experts" in various tasks (Snow et al., 2008). Furthermore, by distributing the workload among multiple annotators, labeling tasks can be completed in a significantly smaller amount of time and it becomes possible to label large datasets efficiently.

From a more general viewpoint, the concept of crowdsourcing goes beyond dedicated platforms such as AMT and often surfaces in more implicit ways. For example, the Web, through its social nature, also exploits the wisdom of crowds to annotate large collections of data. By categorizing texts, tagging images, rating products or clicking links, Web users are generating large volumes of labeled content.

From another perspective, there are tasks for which ground truth labels simply cannot be obtained due to their highly subjective nature. Consider for instance the tasks of sentiment analysis, movie rating or keyphrase extraction. These tasks are subjective in nature and hence no absolute gold standard can be defined. In such cases the only attainable goal is to build a model that captures the wisdom of the crowds (Surowiecki, 2004) as well as possible. For such tasks, crowdsourcing platforms like AMT become a natural solution. However, the large amount of labeled data needed to compensate for the heterogeneity of annotators' expertise can rapidly raise its actual cost beyond acceptable values. Since different annotators have different levels of expertise and personal biases, it is essential to account for the uncertainty associated with their labels, and parsimonious solutions need to be designed that are able to deal with such real world constraints (e.g. annotation cost) and heterogeneity.

Even in situations where ground truth can be obtained, it may be too costly. For example, in medical diagnosis, determining whether a patient has cancer may require a biopsy, which is an invasive procedure and thus should only be used as a last resource. On the other hand, it is rather easy for a diagnostician to consult her colleagues for their opinions before making a decision. Therefore, although there is no crowdsourcing involved in this scenario, there are still multiple experts, with different levels of expertise, providing their own (possibly incorrect) opinions, from which machine learning algorithms have to be able to learn from.

For this kind of problems, an obvious solution is to use majority voting. However, majority voting relies on the frequently wrong assumption that all annotators are equally reliable. Such an assumption is particularly threatening in more heterogeneous environments like AMT, where the reliability of the annotators can vary dramatically (Sheng et al., 2008; Callison-Burch and Dredze, 2010). It is therefore clear that targeted approaches for multiple-annotator settings are required.

So far we have been discussing the use of labels provided by multiple annotators and crowds as a noisy proxy for true outputs y in supervised learning settings. As we discussed, there are numerous factors that make the use this alternative very appealing to machine learning researchers and practitioners, such as cost, efficiency, accessibility, dataset sizes or task subjectiveness. However, there are other ways of

[1] http://www.mturk.com
[2] http://www.crowdflower.com

exploiting data generated by crowds in machine learning tasks. Namely, we will now consider the use of crowdsourced data as additional input features to supervised machine learning algorithms. In order to do so, we shall focus on the particular problem of understanding urban mobility.

During the last decade, the amount of sensory data available for many cities in the world has reached a level which allows for the "pulse" of the city to be accurately captured in real-time. In this data-driven world, technologies that enable high quality and high resolution spatial and temporal data, such as GPS, WiFi, Bluetooth, RFID and NFC, play a vital role. These technologies have become ubiquitous - we can find their use in transit smartcards, toll collection systems, floating car data, fleet management systems, car counters, mobile phones, wearable devices, etc. All this data therefore allows for an unprecedented understanding of how cities behave and how people move within them. However, while this data has the potential of monitoring urban mobility in real-time, it has limited capability on explaining why certain patterns occur. Unfortunately, without a proper understanding of what causes people to move within the city, it also becomes very difficult to make predictions about their mobility, even for a very near future.

Let us consider the case of public transportation. For environmental and societal reasons, public transport has a key role in the future of our cities. However, the challenge of tuning public transport supply adequately to the demand is known to be complicated. While typical planning approaches rely on understanding habitual behavior (Krygsman et al., 2004), it is often found that our cities are too dynamic and difficult to predict. A particularly disruptive case is with special events, like concerts, sports games, sales, festivals or exhibitions (Kwon et al., 2006). Although these are usually planned well in advance, their impact is difficult to predict, even when organizers and transportation operators coordinate. The problem highly increases when several events happen concurrently. To solve these problems, costly processes, heavily reliant on manual search and personal experience, are usual practice in large cities like Singapore, London or Tokyo.

Fortunately, another pervasive technology exists: the internet, which is rich in crowd-generated contextual information. In the internet, users share information about upcoming public special events, comment about their favorite sports teams and artists, announce their likes and dislikes, post what they are doing and much more. Therefore, within this crowdsourced data lay explanations for many of the mobility patterns that we observe. However, even with access to this data, understanding and predicting impacts of future events is not a humanly simple task, as there are many dimensions involved. One needs to consider details such as the type of a public event, popularity of the event protagonists, size of the venue, price, time of day and still account for routine demand behavior, as well as the effect of other co-occurring events. In other words, besides data, sound computational methodologies are also necessary to solve this multidimensional problem.

The combination of the generalized use of smartcard technologies in public transportation systems with the ubiquitous reference to public special events on the internet effectively proposes a potential solution to these limitations. However, the question of developing an efficient and accurate transportation demand model for special event scenarios, in order to predict future demands and to understand the impact of individual events, has remained an unmet challenge. Meeting that challenge would be of great value for public transport operators, regulators and users.

For example, operators can use such information to increase/decrease supply based on the predicted demand and regulators can raise awareness to operators and users on potential non-habitual overcrowding. Furthermore, regulators can also use this information to understand past overcrowding situations, like distinguishing circumstantial from recurrent overcrowding. Lastly, public transport users can enjoy a better service, where there are no disruptions and the supply is adequately adjusted to the demand.

1.2 Contributions

This thesis aims at solving some of the research challenges described in the previous section by proposing novel probabilistic models that make effective use of crowdsourced data for solving machine learning problems. In summary, the main contributions of this thesis are:

- a probabilistic model for supervised learning with multiple annotators where the reliability of the different annotators is treated as a latent variable. The proposed model is capable of distinguishing the good annotators from the less good or even random ones in the absence of ground truth labels, while jointly learning a logistic regression classification model. The particular modeling choice of treating the reliability of the annotators as a latent variable results in various attractive properties, such as the ease of implementation and generalization to other classifiers, the natural extension to structured prediction problems, and the ability to overcome overfitting issues to which more complex models of the annotators expertise can be susceptible as the number of instances labeled per annotator decreases (Rodrigues et al., 2013a).

- a probabilistic approach for sequence labeling using conditional random fields (CRFs) for scenarios where label sequences from multiple annotators are available but there is no actual ground truth. The proposed approach uses an expectation-maximization (EM) algorithm to jointly learn the CRF model parameters, the reliability of the annotators and the estimated ground truth labels sequences (Rodrigues et al., 2013b).

- a generalization of Gaussian process classifiers to explicitly handle multiple annotators with different levels of expertise. In this way, we are bringing a powerful non-linear Bayesian classifier to multiple-annotator settings. This contrasts with previous works, which usually rely on linear classifiers such as logistic regression models. An approximate inference algorithm using expectation propagation (EP) is developed, which is able to compensate for the different biases and reliabilities among the various annotators, thus obtaining more accurate estimates of the ground truth labels. Furthermore, by exploiting the capability of the proposed model to explicitly handle uncertainty, an active learning methodology is proposed, which allows to further reduce annotation costs by actively choosing which instance should be labeled next and which annotator should label it (Rodrigues et al., 2014).

- two fully generative supervised topic models, one for classification and another for regression problems, that account for the different reliabilities of multiple

annotators and corrects their biases. The proposed models are then capable of jointly modeling the words in documents as arising from a mixture of topics, the latent true labels as a result of the empirical distribution over topics of the documents, and the labels of the multiple annotators as noisy versions of the latent ground truth. By also developing a regression model, we are broadening the spectrum of practical applications of the proposed approach and targeting several important machine learning problems. While most of the previous works in the literature focus on classification problems, the equally important topic of learning regression models from crowds has been studied to a much smaller extent. The proposed model is therefore able to learn, for example, how to predict the rating of movies or the number of stars of a restaurant from the noisy or biased opinions of different people. Furthermore, efficient stochastic variational inference algorithms are developed, which allow both models to scale to very large datasets (Rodrigues et al., 2015b,c).

- a probabilistic model that given a non-habitual overcrowding hotspot, which occurs when the public transport demand is above a predefined threshold, it is able to break down the excess of demand into a set of explanatory components. The proposed model uses information regarding special events (e.g. concerts, sports games, festivals, etc.) mined from the Web and preprocessed through text-analysis techniques in order to construct a list of candidate explanations, and assigns to each individual event a share of the overall observed hotspot size. This model is tested using real data from the public transport system of Singapore, which was kindly provided for the purpose of this study by the Land and Transport Authority (LTA) (Pereira et al., 2014a,b).

- a Bayesian additive model with Gaussian process components that combines smartcard data from public transport with crowd-generated information about events that is continuously mined from the Web. In order to perform inference in the proposed model, an expectation propagation algorithm is developed, which allows us to predict the total number of public transportation trips under special event scenarios, thereby contributing to a more adaptive transportation system. Furthermore, for multiple concurrent events, the proposed algorithm is able to disaggregate gross trip counts into their most likely components related to specific events and routine behavior (e.g. commuting). All this information can be of great value not only for public transport operators and planners, but also for event organizers and public transport users in general. Moreover, it is important to point out the wide applicability of the proposed Bayesian additive framework, which can be adapted to different application domains such as electrical signal disaggregation or source separation (Rodrigues et al., 2015a).

Finally, it is worth noting that the source code of the models developed in the context of this thesis and all the datasets used for evaluating them (with the exception of the public transport dataset from LTA, which is proprietary) have been made publicly available for other researchers and practitioners to use in their own applications and for purposes of comparing different approaches. This includes various datasets collected from Amazon mechanical turk for many different tasks, such as classifying posts, news stories, images and music, or even predicting the sentiment of a text,

the number of stars of a review or the rating of movie. As for the source code, it has been properly documented and made available together with brief user manuals. All this information can be found in: http://amilab.dei.uc.pt/fmpr/

1.3 Thesis structure

As previously mentioned, there are two ways in which crowdsourced data can be used to build predictive models: by using the labels provided by multiple annotators as noisy replacements for the true outputs y when these are hard or expensive to obtain, or by using information provided by crowds as additional input features to the model in order to better understand the mapping from inputs $\mathbf{x}$ to outputs y. As such, this thesis is naturally divided in two parts, each corresponding to one of these two settings. Common to both parts is a background chapter — **Chapter 2**. This chapter provides the necessary background knowledge in probabilistic graphical models, Bayesian inference and parameter estimation, which are at the heart of all the approaches developed throughout the thesis.

Part I of this thesis starts with the development of a new class of probabilistic models for learning from multiple annotators and crowds in **Chapter 3**, which we refer to as latent expertise models. A model based on a logistic regression classifier is first presented and then, taking advantage of the extensibility of the proposed class of latent expertise models, a natural extension is developed to sequence labeling problems with conditional random fields.

Logistic regression models and conditional random fields are both linear models of their inputs. Hence, without resorting to techniques such as the use of basis functions, the applicability of those models can be limited, since they cannot define non-linear decision boundaries in order to distinguish between classes. With that in mind, **Chapter 4** presents an extension of Gaussian process classifiers, which are non-linear and non-parametric classifiers, to multiple annotator settings. Furthermore, by taking advantage of some of the properties of the proposed model, an active learning algorithm is also developed.

In **Chapter 5**, the idea of developing non-linear models for learning from multiple annotators and crowds is taken one step further. Since many tasks for which crowdsourcing is typically used deal with complex high-dimensional data such as images or text, in Chapter 5 two supervised topic models for learning from multiple annotators are proposed: one for classification and another for regression problems.

In **Part II** of this thesis, we turn our attention to the use of crowdsourced data as inputs, and in **Chapter 6** an additive model for explaining non-habitual transport overcrowding is proposed. By making use of crowd-generated data about special events, the proposed model is able to break overcrowding hotspots into the contributions of each individual event.

Although presenting satisfactory results in its particular problem, the model presented in Chapter 6 is a simple linear model of its inputs. **Chapter 7** takes the idea of additive formulations beyond linear models, by presenting a Bayesian additive model with Gaussian process components for improving public transport demand predictions through the inclusion of crowdsourced information regarding special events.

In **Chapter 8**, final conclusions regarding the developed models and the obtained results are drawn, and directions for future work are discussed.

Chapter 2

Graphical models, inference and learning

2.1 Probabilistic graphical models

"As far as the laws of mathematics refer to reality, they are not certain, as far as they are certain, they do not refer to reality."
– Albert Einstein, 1956

Probabilities are at the heart of modern machine learning. Probability theory provides us with a consistent framework for quantifying and manipulating uncertainty, which is caused by limitations in our ability to observe the world, our ability to model it, and possibly even because of its innate nondeterminism (Koller and Friedman, 2009). It is, therefore, essential to account for uncertainty when building models of reality. However, probabilistic models can sometimes be quite complex. Hence, it is important to have a simple and compact manner of expressing them.

Probabilistic graphical models provide an intuitive way of representing the structure of a probabilistic model, which not only gives us insights about the properties of the model, such as conditional independencies, but also helps us design new models. A probabilistic graphical model consists of *nodes*, which represent random variables, and *edges* that express probabilistic relationships between the variables. Graphical models can be either undirected or directed. In the latter, commonly known as *Bayesian networks* (Jensen, 1996), the directionality of the edges is used to convey causal relationships (Pearl, 2014). This thesis will make extensive use of directed graphs and a special type of graphs called *factor graphs*, which generalize both directed and undirected graphs. Factor graphs are useful for solving inference problems and enabling efficient computations.

2.1.1 Bayesian networks

Consider an arbitrary joint distribution $p(a, \mathbf{b}, \mathbf{c})$ over the random variables a, $\mathbf{b} = \{b_n\}_{n=1}^N$ and $\mathbf{c} = \{c_n\}_{n=1}^N$ that we want to model. This joint distribution can be factorized in various ways. For instance, making use of the chain rule (or product rule) of probability, it can be verified that $p(a, \mathbf{b}, \mathbf{c}) = p(a)\, p(\mathbf{b}|a)\, p(\mathbf{c}|\mathbf{b}, a)$ and $p(a, \mathbf{b}, \mathbf{c}) = p(\mathbf{c})\, p(a, \mathbf{b}|\mathbf{c})$ are both equivalently valid factorizations of $p(a, \mathbf{b}, \mathbf{c})$. By linking variables, a probabilistic graphical model specifies how a joint distribution

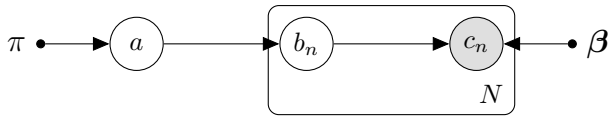


Figure 2.1: Example of a (directed) graphical model.

factorizes. Furthermore, by omitting the links between certain variables, probabilistic graphical models convey a set of conditional independencies, which simplifies the factorization.

Figure 2.1 shows an example of a Bayesian network model representing a factorization of the joint distribution $p(a, \mathbf{b}, \mathbf{c})$. Notice that, instead of writing out the multiple nodes for $\{b_n\}_{n=1}^N$ and $\{c_n\}_{n=1}^N$ explicitly, a rectangle with the label N was used to indicate that the structure within it repeats N times. This rectangle is called a *plate*. Also, we adopted the convention of using large circles to represent random variables (a, b_n and c_n) and small solid circles to denote deterministic parameters (π and $\boldsymbol{\beta}$) (Bishop, 2006). Observed variables are identified by shading their nodes. The unobserved variables, also known as *hidden* or *latent* variables, are indicated using unshaded nodes.

By reading off the dependencies expressed in the probabilistic graphical model of Figure 2.1, the joint distribution of the model, given the parameters π and $\boldsymbol{\beta}$, factorizes as

$$p(a, \mathbf{b}, \mathbf{c}|\pi, \boldsymbol{\beta}) = p(a|\pi) \prod_{n=1}^{N} p(b_n|a)\, p(c_n|b_n, \boldsymbol{\beta}). \tag{2.1}$$

Hence, rather than encoding the probability of every possible assignment to all the variables in the domain, the joint probability breaks down into a product of smaller factors, corresponding to conditional probability distributions over a much smaller space of possibilities, thus leading to a substantially more compact representation that requires significantly less parameters.

So far we have not discussed the form of the individual factors. It turns out that, for *generative models* such as the one in Figure 2.1, a great way to do so is through what is called the *generative process* of the model. Generative models specify how to randomly generate observable data, such as c_n in our example, typically given some latent variables, such as a and b_n. They contrast with *discriminative models* by being full probabilistic models of all the variables, whereas discriminative approaches model only the *target* variables conditional on the observed ones. A generative process is then a description of how to sample observations according to the model.

Returning to our previous example of Figure 2.1, a possible generative process is as follows:[1]

1. Draw $a|\pi \sim \text{Beta}(a|\pi)$
2. For each n
   (a) Draw $b_n|a \sim \text{Bernoulli}(b_n|a)$
   (b) Draw $c_n|b_n, \boldsymbol{\beta} \sim \text{Bernoulli}(c_n|\beta_{b_n})$

[1] A familiar reader might recognize this as an example of a mixture model.

Given this generative process, we know that, for example, the variable a follows a beta distribution[2] with parameter π. Similarly, the conditional probability of c_n given b_n is a Bernoulli distribution with parameter β_{b_n}. Generative processes are then an excellent way of presenting a generative model, and they complement the framework of probabilistic graphical models by conveying additional details. Also, when designing models of reality, it is often useful to think generatively and describe how the observed data came to be. Hence, we shall make extensive use of generative processes throughout this thesis for presenting models.

2.1.2 Factor graphs

Directed and undirected probabilistic graphical models allow a global function of several variables to be expressed as a product of factors over subsets of those variables. These factors can be, for example, probability distributions, as we saw with Bayesian networks (see Eq. 2.1). Factor graphs (Kschischang et al., 2001) differ from directed and undirected graphical models by introducing additional nodes for explicitly representing the factors, which allows them to represent a wider spectrum of distributions (Koller and Friedman, 2009). Figure 2.2 shows an example of a factor graph over the variables a, b, c and d, where the factors are represented using small solid squares.

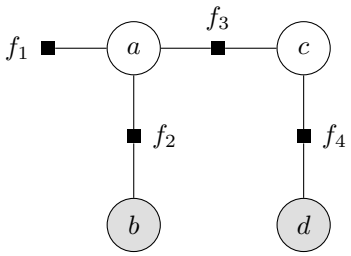


Figure 2.2: Example of a factor graph.

Like Bayesian networks, factor graphs encode a joint probability distribution over a set of variables. However, in factor graphs, the factors do not need to be probability distributions. For example, the factor graph in Figure 2.2 encodes the following factorization of the joint probability distribution over the variables a, b, c and d

$$p(a, b, c, d) = \frac{1}{Z} f_1(a)\, f_2(a, b)\, f_3(a, c)\, f_4(c, d). \tag{2.2}$$

Notice how the factors are now arbitrary functions of subsets of variables. Hence, a normalization constant Z is required to guarantee that the joint distribution $p(a, b, c, d)$ is properly normalized. If the factors correspond to normalized probability distributions, the normalization constant Z can be ignored.

As we shall see later, a great advantage of factor graphs is that they allow the development of efficient inference algorithms by propagating *messages* in the graph (Kschischang et al., 2001; Murphy, 2012) (see Section 2.2.3).

[2] A brief overview of the probability distributions used in this thesis is provided in Appendix A.

2.2 Bayesian inference

Having specified the probabilistic model, the next step is to perform *inference*. Inference is the procedure that allows us to answer various types of questions about the data being modeled, by computing the posterior distribution of the latent variables given the observed ones. For instance, in the example of Figure 2.1, we would like to compute the posterior distribution of a and $\mathbf{b}$, given the observations $\mathbf{c}$. Bayesian inference is a particular method for performing statistical inference, in which Bayes' rule is used to update the posterior distribution of a certain variable(s) as new evidence is acquired.

Bayesian inference can be exact or approximate. In this thesis we will make use of both exact inference and approximate inference procedures, namely *variational inference* (Jordan et al., 1999; Wainwright and Jordan, 2008) and *expectation propagation* (EP) (Minka, 2001).

2.2.1 Exact inference

Without loss of generality, let $\mathbf{z} = \{z_m\}_{m=1}^M$ denote the set of latent variables in a given model, and let $\mathbf{x} = \{x_n\}_{n=1}^N$ denote the observations. Using Bayes' rule, the posterior distribution of $\mathbf{z}$ can be computed as

$$\overbrace{p(\mathbf{z}|\mathbf{x})}^{\text{posterior}} = \frac{p(\mathbf{x},\mathbf{z})}{p(\mathbf{x})} = \frac{\overbrace{p(\mathbf{x}|\mathbf{z})}^{\text{likelihood}}\overbrace{p(\mathbf{z})}^{\text{prior}}}{\underbrace{p(\mathbf{x})}_{\text{evidence}}}. \tag{2.3}$$

The model evidence, or *marginal likelihood*, can be computed by making use of the sum rule of probability to give

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{x}|\mathbf{z})\, p(\mathbf{z}), \tag{2.4}$$

where the summation is replaced by integration in the case that $\mathbf{z}$ is continuous instead of discrete.

At this point, it is important to introduce a broad class of probability distributions called the *exponential family* (Duda and Hart, 1973; Bernardo and Smith, 2009). A distribution over $\mathbf{z}$ with parameters $\boldsymbol{\eta}$ is a member of the exponential family if it can be written in the form

$$p(\mathbf{z}|\boldsymbol{\eta}) = \frac{1}{Z(\boldsymbol{\eta})}\, h(\mathbf{z})\, \exp(\boldsymbol{\eta}^{\mathrm{T}}\mathbf{u}(\mathbf{z})), \tag{2.5}$$

where $\boldsymbol{\eta}$ are called the *natural parameters*, $\mathbf{u}(\mathbf{z})$ is a vector of *sufficient statistics* and $h(\mathbf{z})$ is a scaling constant, often equal to 1. The normalization constant $Z(\boldsymbol{\eta})$, also called the *partition function*, ensures that the distribution is normalized.

Many popular distributions belong to the exponential family, such as the Gaussian, exponential, beta, Dirichlet, Bernoulli, multinomial and Poisson (Bernardo and Smith, 2009). Exponential family members have many interesting properties, which make them so appealing for modelling random variables. For example, the exponential family has finite-sized sufficient statistics, which means that the data can be compressed into a fixed-sized summary without loss of information.

A particularly useful property of exponential family members is that they are closed under multiplication. This means that if we multiply together two exponential family distributions $p(\mathbf{z})$ and $p(\mathbf{z}')$, the product $p(\mathbf{z}, \mathbf{z}') = p(\mathbf{z})\, p(\mathbf{z}')$ will also be in the exponential family. This property is closely related to the concept of *conjugate priors*. In general, for a given posterior distribution $p(\mathbf{z}|\mathbf{x})$, we seek a prior distribution $p(\mathbf{z})$ so that when multiplied by the likelihood $p(\mathbf{x}|\mathbf{z})$, the posterior has the same functional form as the prior. This is called a conjugate prior. For any member of the exponential family there exists a conjugate prior (Bishop, 2006; Bernardo and Smith, 2009). For example, the conjugate prior for the parameters of a multinomial distribution is the Dirichlet distribution, while the conjugate prior for the mean of a Gaussian is another Gaussian. As we shall see, the choice of conjugate priors greatly simplifies the calculations involved in Bayesian inference. Furthermore, the fact that the posterior keeps the same functional form as the prior, allows the development of online learning algorithms, where the posterior is used as the new prior, as new observations are sequentially acquired.

2.2.2 Variational inference

Unfortunately, for various models of practical interest, it is infeasible to evaluate the posterior distribution exactly or to compute expectations with respect to it. There are several reasons for this. For example, it might be the case where the dimensionality of the latent space is too high to work with directly, or because the form of the posterior distribution is so complex that computing expectations is not analytically tractable, or even because some of the required integrations might not have closed-form solutions. Consider, for example, the case of the model of Figure 2.1. The posterior distribution over the latent variables a and $\mathbf{b}$ is given by

$$p(a, \mathbf{b}|\mathbf{c}) = \frac{p(a, \mathbf{b}, \mathbf{c})}{p(\mathbf{c})} = \frac{p(a|\pi) \prod_{n=1}^{N} p(b_n|a)\, p(c_n|b_n, \boldsymbol{\beta})}{\int_a \sum_{\mathbf{b}} p(a|\pi) \prod_{n=1}^{N} p(b_n|a)\, p(c_n|b_n, \boldsymbol{\beta})}. \tag{2.6}$$

The numerator can be easily evaluated for any combination of the latent variables, but the denominator is intractable to compute. In such cases, where computing the exact posterior distribution is infeasible, we need to resort to approximate inference algorithms, which turn the computation of posterior distributions into a tractable problem, by trading off computation time for accuracy.

We can differentiate between two major classes of approximate inference algorithms, depending on whether they rely on stochastic or deterministic approximations. Stochastic techniques for approximate inference, such as Markov chain Monte Carlo (MCMC) (Gilks, 2005), rely on sampling and have the property that given infinite computational resources they can generate exact results. For example, MCMC methods are based on Monte Carlo approximations, whose main idea is to use repeated sampling to approximate the desired distribution. MCMC methods iteratively construct a Markov chain of samples, which, at the some point, converges. At this stage, the sample draws are close to the true posterior distribution and they can be collected to approximate the required expectations. However, in practice, it is hard to determine when a chain has converged or "mixed". Furthermore, the number of samples required for the chain to mix can be very large. As a consequence, MCMC methods tend to be computationally demanding, which generally restricts

their application to small-scale problems (Bishop, 2006). On the other hand, deterministic methods, such as variational inference and expectation propagation, are based on analytical approximations to the posterior distribution. Therefore, they tend to scale better to large-scale inference problems, making them better suited for the models proposed in this thesis.

Variational inference, or *variational Bayes* (Jordan et al., 1999; Wainwright and Jordan, 2008), constructs an approximation to the true posterior distribution $p(\mathbf{z}|\mathbf{x})$ by considering a family of tractable distributions $q(\mathbf{z})$. A tractable family can be obtained by relaxing some constraints in the true distribution. Then, the inference problem is to optimize the parameters of the new distribution so that the approximation becomes as close as possible to the true posterior. This reduces inference to an optimization problem.

The closeness between the approximate posterior $q(\mathbf{z})$, known as the *variational distribution*, and the true posterior $p(\mathbf{z}|\mathbf{x})$ can be measured by the Kullback-Leibler (KL) divergence (MacKay, 2003), which is given by

$$\mathbb{KL}(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})) = \int_{\mathbf{z}} q(\mathbf{z}) \log \frac{q(\mathbf{z})}{p(\mathbf{z}|\mathbf{x})}. \tag{2.7}$$

Notice that the KL divergence is an asymmetric measure. Hence, we could have chosen the reverse KL divergence, $\mathbb{KL}(p(\mathbf{z}|\mathbf{x})||q(\mathbf{z}))$, but that would require us to be able to take expectations with respect to $p(\mathbf{z}|\mathbf{x})$. In fact, that would lead to a different kind of approximation algorithm, called expectation propagation, which shall be discussed in Section 2.2.3.

Unfortunately, the KL divergence in (2.7) cannot be minimized directly. However, we can find a function that we can minimize, which is equal to it up to an additive constant, as follows

$$\begin{aligned}
\mathbb{KL}(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})) &= \mathbb{E}_q\left[\log \frac{q(\mathbf{z})}{p(\mathbf{z}|\mathbf{x})}\right] \\
&= \mathbb{E}_q[\log q(\mathbf{z})] - \mathbb{E}_q[\log p(\mathbf{z}|\mathbf{x})] \\
&= \mathbb{E}_q[\log q(\mathbf{z})] - \mathbb{E}_q\left[\log \frac{p(\mathbf{z},\mathbf{x})}{p(\mathbf{x})}\right] \\
&= -\underbrace{(\mathbb{E}_q[\log p(\mathbf{z},\mathbf{x})] - \mathbb{E}_q[\log q(\mathbf{z})])}_{\mathcal{L}(q)} + \underbrace{\log p(\mathbf{x})}_{\text{const.}}.
\end{aligned} \tag{2.8}$$

The $\log p(\mathbf{x})$ term of (2.8) does not depend on q and thus it can be ignored. Minimizing the KL divergence is then equivalent to maximizing $\mathcal{L}(q)$, which is called the *evidence lower bound*. The fact that $\mathcal{L}(q)$ is a lower bound on the log model evidence, $\log p(\mathbf{x})$, can be emphasized by recalling Jensen's inequality to notice that, due to the concavity of the logarithmic function, $\log \mathbb{E}[p(\mathbf{x})] \geqslant \mathbb{E}[\log p(\mathbf{x})]$. Thus,

Jensen's inequality can be applied to the logarithm of the model evidence to give

$$\begin{aligned}\log p(\mathbf{x}) &= \log \int_{\mathbf{z}} p(\mathbf{z}, \mathbf{x}) \\ &= \log \int_{\mathbf{z}} \frac{q(\mathbf{z})}{q(\mathbf{z})} p(\mathbf{z}, \mathbf{x}) \\ &= \log \mathbb{E}_q \left[\frac{p(\mathbf{z}, \mathbf{x})}{q(\mathbf{z})} \right] \\ &\geqslant \underbrace{\mathbb{E}_q[\log p(\mathbf{z}, \mathbf{x})] - \mathbb{E}_q[\log q(\mathbf{z})]}_{\mathcal{L}(q)}. \end{aligned} \tag{2.9}$$

The evidence lower bound $\mathcal{L}(q)$ is tight when $q(\mathbf{z}) \approx p(\mathbf{z}|\mathbf{x})$, in which case $\mathcal{L}(q) \approx \log p(\mathbf{x})$. The goal of variational inference is then to find the parameters of the variational distribution $q(\mathbf{z})$, known as the *variational parameters*, that maximize the evidence lower bound $\mathcal{L}(q)$.

The key to make variational inference work is to find a tractable family of approximate distributions $q(\mathbf{z})$ for which the expectations in (2.9) can be easily computed. The most common choice for $q(\mathbf{z})$ is a fully factorized distribution, such that $q(\mathbf{z}) = \prod_{m=1}^{M} q(z_m)$. This is called a *mean-field* approximation. In fact, mean field theory is by itself a very important topic in statistical physics (Parisi, 1988).

Using a mean-field approximation corresponds to assuming that the latent variables $\{z_i\}_{i=1}^{M}$ are independent of each other. Hence, the expectations in (2.9) become sums of simpler expectations. For example, the term $\mathbb{E}_q[\log q(\mathbf{z})]$ becomes $\mathbb{E}_q[\log q(\mathbf{z})] = \sum_{m=1}^{M} \mathbb{E}_q[\log q(z_m)]$. The evidence lower bound, $\mathcal{L}(q)$, can then be optimized by using a coordinate ascent algorithm that iteratively optimizes the variational parameters of the approximate posterior distribution of each latent variable $q(z_m)$ in turn, holding the others fixed, until a convergence criterium is met. This ensures convergence to a local maximum of $\mathcal{L}(q)$. We shall see practical examples of variational inference in Chapter 5.

2.2.3 Expectation propagation

Expectation propagation (EP) (Minka, 2001) is another deterministic method for approximate inference. It differs from variational inference by considering the reverse KL divergence $\mathbb{KL}(p||q)$ instead of $\mathbb{KL}(q||p)$. This gives the approximation different properties.

Consider an arbitrary probabilistic graphical model encoding a joint probability distribution over observations $\mathbf{x} = \{x_n\}_{n=1}^{N}$ and latent variables $\mathbf{z} = \{z_m\}_{m=1}^{M}$, so that it factorizes as a product of factors $f_i(\mathbf{z})$

$$p(\mathbf{z}, \mathbf{x}) = \prod_i f_i(\mathbf{z}), \tag{2.10}$$

where we omitted the dependence of the factors on the observations for the ease of exposition and to keep the presentation coherent with the literature (Minka, 2001; Bishop, 2006; Murphy, 2012). The posterior distribution of the latent variables is then given by

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{z}, \mathbf{x})}{p(\mathbf{x})} = \frac{1}{p(\mathbf{x})} \prod_i f_i(\mathbf{z}). \tag{2.11}$$

The model evidence $p(\mathbf{x})$ is obtained by marginalizing over the latent variables, i.e. $p(\mathbf{x}) = \int_{\mathbf{z}} \prod_i f_i(\mathbf{z})$, where the integral is replaced by a summation in the case that $\mathbf{z}$ is discrete. However, without loss of generality, we shall assume for the rest of this section that $\mathbf{z}$ is continuous.

In expectation propagation, we consider an approximation to the posterior distribution of the form

$$q(\mathbf{z}) = \frac{1}{Z_{\text{EP}}} \prod_i \widetilde{f}_i(\mathbf{z}), \tag{2.12}$$

where the normalization constant Z_{EP} is required to ensure that the distribution integrates to unity. Just as with variational inference, the approximate posterior $q(\mathbf{z})$ needs to be restricted in some way, in order for the required computations to be tractable. In particular, we shall assume that the approximate factors $\widetilde{f}_i(\mathbf{z})$ belong to the exponential family, so that the product of all the factors is also in the exponential family and thus can be described by a finite set of sufficient statistics.

As previously mentioned, expectation propagation considers the reverse KL, $\mathbb{KL}(p||q)$. However, minimizing the global KL divergence between the true posterior and the approximation, $\mathbb{KL}(p(\mathbf{z}|\mathbf{x})||q(\mathbf{z}))$, is generally intractable. Alternatively, one could consider minimizing the local KL divergences between the different individual factors, $\mathbb{KL}(f_i(\mathbf{z})||\widetilde{f}_i(\mathbf{z}))$, but that would give no guarantees that the product of all the factors $\prod_i \widetilde{f}_i(\mathbf{z})$ would be a good approximation to $\prod_i f_i(\mathbf{z})$, and actually, in practice, it leads to poor approximations (Bishop, 2006). EP uses a tractable compromise between these two alternatives, where the approximation is made by optimizing each factor in turn in the context of all the remaining factors.

Let us now see in more detail how the posterior approximation of EP is done. Suppose we want to refine the factor approximation $\widetilde{f}_j(\mathbf{z})$, and let $p^{\setminus j}(\mathbf{z})$ and $q^{\setminus j}(\mathbf{z})$ be the product of all the other factors (exact or approximate) that do not involve j, i.e. $p^{\setminus j}(\mathbf{z}) \triangleq \prod_{i \neq j} f_i(\mathbf{z})$ and $q^{\setminus j}(\mathbf{z}) \triangleq \prod_{i \neq j} \widetilde{f}_i(\mathbf{z})$. This defines the *context* of a factor. Ideally, in order to optimize a given factor $\widetilde{f}_j(\mathbf{z})$, we would like to minimize the KL divergence $\mathbb{KL}(p(\mathbf{z}|\mathbf{x})||q(\mathbf{z}))$, which can be written as

$$\mathbb{KL}\left(\frac{1}{p(\mathbf{x})} f_j(\mathbf{z})\, p^{\setminus j}(\mathbf{z}) \middle|\middle| \frac{1}{Z_{\text{EP}}} \widetilde{f}_j(\mathbf{z})\, q^{\setminus j}(\mathbf{z})\right), \tag{2.13}$$

but, as previously mentioned, this is intractable to compute. We can make this tractable by assuming that the approximations we already made, $q^{\setminus j}(\mathbf{z})$, are a good approximation for the rest of the distribution, i.e. $q^{\setminus j}(\mathbf{z}) \approx p^{\setminus j}(\mathbf{z})$. This corresponds to making the approximation of the factor $\widetilde{f}_j(\mathbf{z})$ in the context of all the other factors, which ensures that the approximation is most accurate in the regions of high posterior probability as defined by the remaining factors (Minka, 2001). Of course the closer the context approximation $q^{\setminus j}(\mathbf{z})$ is to the true context $p^{\setminus j}(\mathbf{z})$, the better the approximation for the factor $\widetilde{f}_j(\mathbf{z})$ will be. EP starts by initializing the factors $\widetilde{f}_i(\mathbf{z})$ and then iteratively refines each of these factors one at the time, much like the coordinate ascent algorithm used in variational inference iteratively optimizes the evidence lower bound with respect to one of the variational parameters.

Let $q(\mathbf{z})$ be the current posterior approximation and let $\widetilde{f}_j(\mathbf{z})$ be the factor we wish to refine. The context $q^{\setminus j}(\mathbf{z})$, also known as the *cavity* distribution, can be

obtained either by explicitly multiplying all the other factors except $\widetilde{f}_j(\mathbf{z})$ or, more conveniently, by dividing the current posterior approximation $q(\mathbf{z})$ by $\widetilde{f}_j(\mathbf{z})$

$$q^{\setminus j}(\mathbf{z}) = \frac{q(\mathbf{z})}{\widetilde{f}_j(\mathbf{z})}. \tag{2.14}$$

Notice that $q^{\setminus j}(\mathbf{z})$ corresponds to an unnormalized distribution, so that it requires its own normalization constant Z_j in order to be properly normalized. We then wish to estimate the new approximate distribution $q^{\text{new}}(\mathbf{z})$ that minimizes the KL divergence

$$\mathbb{KL}\left(\frac{1}{Z_j} f_j(\mathbf{z})\, q^{\setminus j}(\mathbf{z}) \middle|\middle| q^{\text{new}}(\mathbf{z})\right). \tag{2.15}$$

It turns out that, as long as $q^{\text{new}}(\mathbf{z})$ is in the exponential family, this KL divergence can be minimized by setting the expected sufficient statistics of $q^{\text{new}}(\mathbf{z})$ to the corresponding moments of $Z_j^{-1} f_j(\mathbf{z})\, q^{\setminus j}(\mathbf{z})$ (Koller and Friedman, 2009; Murphy, 2012), where the normalization constant is given by $Z_j = \int_{\mathbf{z}} f_j(\mathbf{z})\, q^{\setminus j}(\mathbf{z})$. The revised factor can then be computed as

$$\widetilde{f}_j(\mathbf{z}) = Z_j \frac{q^{\text{new}}(\mathbf{z})}{q^{\setminus j}(\mathbf{z})}. \tag{2.16}$$

In many situations, it is useful to interpret the expectation propagation algorithm as message-passing in a factor graph. This perspective can be obtained by viewing the approximation $\widetilde{f}_j(\mathbf{z})$ as the message that factor j sends to the rest of the network, and the context $q^{\setminus j}(\mathbf{z})$ as the collection of messages that factor j receives. The algorithm then alternates between computing expected sufficient statistics and propagating these in the graph, hence the name "expectation propagation".

By considering the reverse KL divergence, the approximations produced by EP have rather different properties than those produced by variational inference. Namely, while the former are "moment matching", the latter are "mode seeking". This is particularly important when the posterior is highly multimodal. Multimodality can be caused by non-identifiability in the latent space or by complex nonlinear dependencies (Bishop, 2006). When a multimodal distribution is approximated by a unimodal one using the KL divergence $\mathbb{KL}(q||p)$, the resulting approximation will fit one of the modes. Conversely, if we use the reverse KL divergence, $\mathbb{KL}(p||q)$, the approximation obtained would average across all the modes. Hence, depending on the practical approximation at hand, one approach is preferable over the other. We shall see practical applications of EP in Chapters 4 and 7.

2.3 Parameter estimation

A probabilistic model usually consists of variables, relationships between variables, and parameters. Parameters differ from latent variables by being single-valued instead of having a probability distribution over a range of possible values associated. Section 2.2 described exact and approximate methods for inferring the posterior distribution of the latent variables given the observed ones. In this section, we will give an overview of common approaches to find point-estimates for the parameters of a model, that will be useful for the models proposed in this thesis.

2.3.1 Maximum likelihood and MAP

Let $\mathbf{x} = \{x_n\}_{n=1}^N$ be a set of observed variables and $\boldsymbol{\theta}$ denote the set of model parameters. The most widely known method for determining the values of $\boldsymbol{\theta}$ is *maximum-likelihood* estimation (MLE). As the name suggests, it consists of setting the parameters $\boldsymbol{\theta}$ to the values that maximize the likelihood of the observations. For both computational and numerical stability reasons, it is convenient to maximize the logarithm of likelihood. The maximum-likelihood estimator is then given by

$$\boldsymbol{\theta}_{\text{MLE}} = \arg\max_{\boldsymbol{\theta}} \big(\log p(\mathbf{x}|\boldsymbol{\theta}) \big). \tag{2.17}$$

This maximization problem can be easily solved by taking derivatives of $\log p(\mathbf{x}|\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$ and equating them to zero in order to obtain a solution.

In many situations, we want to incorporate prior knowledge regarding the parameters $\boldsymbol{\theta}$. This can be done by defining a prior distribution over $\boldsymbol{\theta}$, $p(\boldsymbol{\theta})$. This can be useful, for instance, for regularization purposes. Suppose $\boldsymbol{\theta}$ corresponds to a vector of weights. We can prevent these to be arbitrarily large, by assigning $\boldsymbol{\theta}$ a Gaussian prior with a small variance. In fact, as it turns out, this corresponds to a popular type of regularization called ℓ_2-regularization (see Ng (2004) for an insightful discussion on different types of regularization).

Given a prior distribution over the parameters, $p(\boldsymbol{\theta})$, the posterior distribution can be obtained by making use of Bayes' theorem

$$p(\boldsymbol{\theta}|\mathbf{x}) = \frac{p(\mathbf{x}|\boldsymbol{\theta})\, p(\boldsymbol{\theta})}{p(\mathbf{x})}. \tag{2.18}$$

Since $p(\mathbf{x})$ is constant w.r.t. $\boldsymbol{\theta}$, we can find a point-estimate for the parameters by maximizing the logarithm of numerator

$$\boldsymbol{\theta}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} \big(\log p(\mathbf{x}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) \big). \tag{2.19}$$

This is called a *maximum-a-posteriori* (MAP) estimate. Notice that, contrarily to Bayesian inference, where the full posterior distribution is computed, MAP estimation determines a single point-estimate for the parameters $\boldsymbol{\theta}$, which corresponds to the mode of the posterior distribution.

2.3.2 Expectation maximization

Many models of practical interest often combine observed variables with latent ones. Let $\mathbf{z} = \{z_m\}_{m=1}^M$ denote the set of latent variables in the model. Without loss of generality, we shall assume that $\mathbf{z}$ is discrete. However, the discussion would still apply if $\mathbf{z}$ was continuous, simply by replacing the summations over $\mathbf{z}$ by integrals.

Ideally, for models with unobserved variables, we would like to find the parameters $\boldsymbol{\theta}$ that maximize the (log) marginal likelihood of the data (or model evidence) given by

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \log \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}). \tag{2.20}$$

Unfortunately, this is generally intractable to maximize directly because of the summation that appears inside the logarithm and prevents it from acting directly on

the joint distribution, which would allow us to exploit the factorization of the latter to re-write $\log p(\mathbf{x}|\boldsymbol{\theta})$ as a sum of logarithms of simpler and more tractable terms. Furthermore, this optimization problem is not convex, which makes it even harder to solve.

On the other hand, if the latent variables $\mathbf{z}$ were observed, then we could simply find the parameters $\boldsymbol{\theta}$ that maximize the *complete-data* log likelihood, $\log p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})$. Since the latent variables are not observed, we cannot maximize $\log p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})$ directly. However, we can instead maximize its expected value under a current estimate, $q(\mathbf{z})$, of the posterior distribution, $p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})$, which is given by

$$\mathbb{E}_q[\log p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})] = \sum_{\mathbf{z}} q(\mathbf{z}) \log p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}). \tag{2.21}$$

Using the newly estimated parameters, we can then revise our estimate, $q(\mathbf{z})$, of the posterior distribution of the latent variables, $p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})$. Iterating between these two steps gives rise to the *expectation-maximization* (EM) algorithm (Dempster et al., 1977).

The EM algorithm is then a general method for estimating the parameters in a probabilistic model in the presence of latent variables. It consists of two steps: the E-step and the M-step. In the E-step, the posterior distribution of the latent variables $q(\mathbf{z}) = p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}^{\text{old}})$ is estimated given some "old" estimate of the parameter values $\boldsymbol{\theta}^{\text{old}}$. In the M-step, we find the "new" parameters $\boldsymbol{\theta}^{\text{new}}$ that maximize

$$\boldsymbol{\theta}^{\text{new}} = \arg\max_{\boldsymbol{\theta}} \left(\sum_{\mathbf{z}} q(\mathbf{z}) \log p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) \right). \tag{2.22}$$

The EM algorithm iterates between these two steps until a convergence criterion is satisfied. At each iteration, the algorithm guarantees that the log likelihood of the observed data, $\log p(\mathbf{x}|\boldsymbol{\theta})$, increases. In order to verify that, let us recall Eq. 2.8 and re-write it as

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \mathbb{E}_q[\log p(\mathbf{z}, \mathbf{x}|\boldsymbol{\theta})] + \mathcal{H}(q) + \mathbb{KL}(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})), \tag{2.23}$$

where we made the model parameters $\boldsymbol{\theta}$ explicit, and defined the entropy of q, $\mathcal{H}(q) \triangleq -\mathbb{E}_q[\log q(\mathbf{z})]$. Since the KL divergence is always non-negative, we have that

$$\log p(\mathbf{x}|\boldsymbol{\theta}) \geqslant \mathbb{E}_q[\log p(\mathbf{z}, \mathbf{x}|\boldsymbol{\theta})] + \mathcal{H}(q). \tag{2.24}$$

Hence, the right-hand side of (2.24) is a lower-bound on the log marginal likelihood $\log p(\mathbf{x}|\boldsymbol{\theta})$. This bound is tight when the KL divergence term vanishes from (2.23). The KL divergence $\mathbb{KL}(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}))$ is zero when $q(\mathbf{z}) = p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})$. Hence, the E-step of the EM algorithm makes this bound tight. When this bound is tight, we have that $\log p(\mathbf{x}|\boldsymbol{\theta})$ is equal to $\mathbb{E}_q[\log p(\mathbf{z}, \mathbf{x}|\boldsymbol{\theta})]$ up to an additive constant, $\mathcal{H}(q)$, which does not depend on the model parameters $\boldsymbol{\theta}$ (see Eq. 2.24). The expected complete-data log likelihood, $\mathbb{E}_q[\log p(\mathbf{z}, \mathbf{x}|\boldsymbol{\theta})]$, can then be used as a proxy for optimizing $\boldsymbol{\theta}$, which corresponds to the M-step of the EM algorithm.

This view of EM as optimizing a lower bound on the (log) marginal likelihood of the data highlights its close relation with variational inference. In fact, when the exact posterior over the latent variables, $p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})$, is intractable to compute, variational inference can be used in the E-step to approximate it. This procedure is commonly known as *variational Bayes* EM (VBEM) (Bernardo et al., 2003; Murphy, 2012).

Part I

Learning from crowds

Chapter 3

Latent expertise models

3.1 Introduction

"I think it is much more interesting to live with uncertainty than to live with answers that might be wrong."
– Richard Feynman

As explained in Chapter 1, learning from multiple annotators is an increasingly important research topic in modern machine learning. The main intuition is that different annotators have different levels of expertise. As it turns out, these differences in expertise can have significant impacts in practical applications, as various authors have shown (e.g. Snow et al. (2008); Raykar et al. (2010)). It is therefore essential to account for their effect when learning predictive models using data labeled by multiple annotators and crowds.

Let $\mathcal{D} = \{\mathbf{x}_n, c_n\}_{n=1}^N$ be a dataset of size N, where for each input vector $\mathbf{x}_n \in \mathbb{R}^D$ we are given the corresponding correct target c_n. In a typical supervised learning setting, our goal is to find a function that maps inputs $\mathbf{x}_n$ to targets c_n, such that the prediction error on unseen inputs $\mathbf{x}_*$ is minimized. When learning from multiple annotators, instead of a single ground truth label c_n, for each instance $\mathbf{x}_n$ we are given a set of labels $\mathbf{y}_n = \{y_n^1, \ldots, y_n^R\}$, which correspond to the noisy answers of multiple annotators. In such cases, the simplest solution would be to use majority voting to estimate c_n from $\mathbf{y}_n$, or to use the average of $\mathbf{y}_n$ if the answers y_n^r are continuous variables. This corresponds to assuming that all the R annotators are equally reliable, since their votes are weighted equally. Unfortunately, this assumption generally does not hold in practice (Snow et al., 2008; Sheng et al., 2008; Callison-Burch, 2009).

Instead of relying on majority voting, Dawid and Skene (1979) proposed an approach for estimating the error rates of multiple patients (annotators) given their responses (labels) to multiple medical questions (instances). The idea behind this approach, and many others that later followed, is to consider that there is an unobserved or latent ground truth c_n. The different annotators are then assumed to provide noisy versions y_n^r of this latent ground truth, such that the annotators will provide the correct label c_n with probability $p(y_n^r = c_n|c_n)$, or some other (incorrect) label l with probability $p(y_n^r = l|c_n)$. Notice that these probabilities are conditioned on the latent true class c_n. Hence, each annotator can have different probabilities of providing a correct label depending on what the latent true class c_n is. For the

sake of simplicity and without loss of generality, let us consider for now that the responses are binary variables, such that $y_n^r \in \{0, 1\}$. Translating this idea into a probabilistic model, yields the following generative process:

1. For each question n
   (a) For each patient r
      i. If true class $c_n = 1$
         Draw patient's answer $y_n^r|\alpha^r \sim \text{Bernoulli}(y_n^r|\alpha^r)$
      ii. If true class $c_n = 0$
         Draw patient's answer $y_n^r|\beta^r \sim \text{Bernoulli}(y_n^r|1 - \beta^r)$

Since we are focusing on binary classification problems, the parameters of the Bernoullis, α^r and β^r, can be interpreted as the sensitivity and specificity, respectively, of the r^{th} annotator. Figure 3.1 shows the corresponding probabilistic graphical model, where N denotes the number of instances (questions) and R is the total number of annotators (patients). Notice how the ground truth labels c_n are represented using an unshaded circle, indicating that they are latent variables. If the sensitivities $\boldsymbol{\alpha} = \{\alpha^r\}_{r=1}^R$ and specificities $\boldsymbol{\beta} = \{\beta^r\}_{r=1}^R$ of the different annotators were known, it would be easy to estimate the ground truth $\mathbf{c} = \{c_n\}_{n=1}^N$. Similarly, if the ground truth $\mathbf{c}$ was known, it would be straightforward to estimate the sensitivities $\boldsymbol{\alpha}$ and specificities $\boldsymbol{\beta}$. This apparent chicken-and-the-egg problem can be solved by using an EM algorithm, as proposed by Dawid and Skene (1979).

Although this work just focuses on estimating the hidden ground truth labels and the error rates of the different annotators, it inspired other works where there is an explicit attempt to learn a classifier. For example, Smyth et al. (1995) proposed a similar approach to solve the problem of volcano detection and classification in Venus imagery with data labeled by multiple experts. As in previous works, this approach relies on a latent variable model where the ground truth labels are treated as latent variables. The main difference is that the authors use the estimated (probabilistic) ground truth labels to explicitly learn a classifier.

More recently, Snow et al. (2008) demonstrated that learning from labels provided by multiple non-expert annotators can be as good as learning from the labels of one expert. Such kind of findings inspired the development of new approaches that, unlike previous ones (Smyth et al., 1995; Donmez and Carbonell, 2008; Sheng et al., 2008), do not rely on repeated labeling, i.e. having the same annotators labeling the same set of instances. This is the case of the approach proposed by Raykar et al. (2009, 2010), in which the reliabilities of the different annotators and a classifier are learnt jointly. The idea is to extend Dawid and Skene's framework

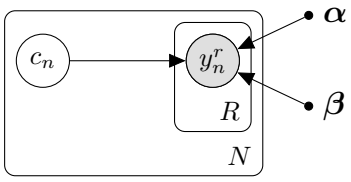


Figure 3.1: Graphical model of the approach of Dawid and Skene (1979).

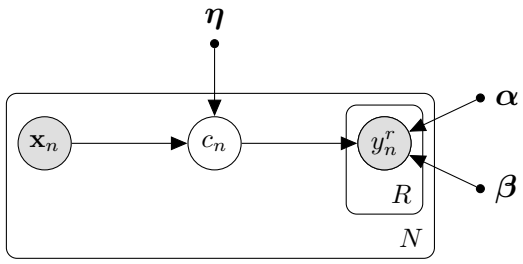


Figure 3.2: Graphical model of the approach of Raykar et al. (2010).

by modeling the ground truth labels with a logistic regression parameterized by a vector of coefficients $\boldsymbol{\eta}$. The generative process is then as follows:

1. For each instance $\mathbf{x}_n$
   (a) Draw true class $c_n|\mathbf{x}_n, \boldsymbol{\eta} \sim \text{Bernoulli}(c_n|\text{Sigmoid}(\mathbf{x}_n, \boldsymbol{\eta}))$
   (b) For each annotator r
      i. If true class $c_n = 1$
         Draw annotator's answer $y_n^r|\alpha^r \sim \text{Bernoulli}(y_n^r|\alpha^r)$
      ii. If true class $c_n = 0$
         Draw annotator's answer $y_n^r|\beta^r \sim \text{Bernoulli}(y_n^r|1-\beta^r)$

In the generative process above, the notation $\text{Sigmoid}(\mathbf{x}_n, \boldsymbol{\eta})$ is used to denote a logistic sigmoid function, which is commonly used in logistic regression and is defined as: $\text{Sigmoid}(\mathbf{x}_n, \boldsymbol{\eta}) \triangleq 1/(1 + \exp(-\boldsymbol{\eta}^{\mathrm{T}}\mathbf{x}_n))$. Figure 3.2 shows the corresponding graphical model representation.

By using a logistic regression to model the ground truth labels as a function of the inputs, the approach proposed by Raykar et al. is able to generalize across the instances labeled by the different annotators. Hence, they are not required to label exactly the same instances, thus allowing the dataset to be split and distributed among different annotators for labeling. This can be achieved by replacing R by R_n in the graphical model, where R_n denotes the annotators that labeled the n^{th} instance, and making changes in the equations accordingly. Nevertheless, for the ease of exposition, we shall assume that, for all the models in this thesis, all annotators label all the instances, i.e. $R_n = R$. However, this is for presentation purposes only, since all the implementations take this into consideration. As with previous approaches, the authors use an EM algorithm to infer the latent ground truth labels $\mathbf{c}$ (E-step), as well as to estimate the sensitivities $\boldsymbol{\alpha}$ and specificities $\boldsymbol{\beta}$ of the annotators and the coefficients $\boldsymbol{\eta}$ of the logistic regression (M-step).

The model proposed by Raykar et al. (2010) relies on the assumption that the labels provided by the different annotators do not dependent on the instances that they are labeling. In other words, it assumes the conditional independence of y_n^r on $\mathbf{x}_n$ given c_n, which we can readily observe from the graphical model in Figure 3.2, by noticing that once the true classes c_n are observed, the annotators labels y_n^r become independent of $\mathbf{x}_n$ (they only depend on c_n). This is reasonable for a wide class of practical applications. Nevertheless, Yan et al. (2010) question this assumption and relax it by proposing a model where the annotators' reliabilities are conditioned

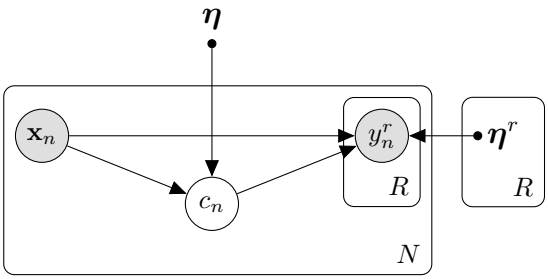


Figure 3.3: Graphical model of the approach of Yan et al. (2010).

on the input instance through a second logistic regression with annotator-specific coefficients $\boldsymbol{\eta}^r$. Figure 3.3 shows the graphical model proposed by Yan et al. (2010). In fact, this line of work inspired many interesting extensions. For example, Yan et al. (2011) suggest an active learning methodology for reducing the annotation costs, by selecting which instance should be labeled next and which annotator should label it. Yet, with the concern that the reliability of the annotators may change over time, Donmez et al. (2010) propose the use of a particle filter (Bishop, 2006) to model their time-varying accuracies.

So far we have been discussing approaches that were developed mainly for classification problems. However, although to a smaller extent, there are other research works that address different types of target variables as well. For example, from a regression perspective, the problem of learning from multiple annotators has been addressed in the context of Gaussian processes by Groot et al. (2011). For ranking problems, the problem of learning from multiple annotators has been studied, for example, by Wu et al. (2011).

Regardless of the nature of the target variables, the approaches described above all share one common aspect: they are centered on the unobserved true labels, of which the various annotators are assumed to provide noisy versions. Therefore, they typically approach the problem of learning from multiple annotators by treating the unobserved true labels as latent variables in a probabilistic framework, which hinders a natural extension to structured prediction problems such as sequence labeling tasks, due to the combinatorial explosion of possible outcomes of the latent variables. Sequence labeling refers to the supervised learning task of assigning a label to each element of a sequence. Typical examples are part-of-speech (POS) tagging, named entity recognition (NER) and gene prediction (Allen et al., 2004; Allen and Salzberg, 2005). In such tasks, the individual labels cannot be considered as detached from the context, i.e. the preceding and succeeding elements of the sequence and their corresponding labels.

Two of the most popular sequence models are hidden Markov models (HMM) (Rabiner, 1990) and conditional random fields (CRF) (Lafferty et al., 2001). Due to the usual high-dimensional feature spaces (especially considering CRFs), these models frequently require large amounts of labeled data to be properly trained, which complicates the construction and release of datasets and makes it almost prohibitive to do with a single annotator. Although in some domains, the use of unlabeled data can help in making this problem less severe (Bellare and Mccallum, 2007), a more natural solution is to rely on multiple annotators. For example, for many tasks, Amazon mechanical turk (AMT) can be used to label large amounts

of sequential data (Callison-Burch and Dredze, 2010). However, the large numbers needed to compensate for the heterogeneity of annotators expertise rapidly raise its actual cost beyond acceptable values. A parsimonious solution needs to be designed that is able to deal with such real world constraints and heterogeneity.

In this chapter, we propose a new class of models for learning from multiple annotators that contrasts with previous works by focusing on the reliability of the different annotators. As we shall see, by treating the information regarding the unknown reliability of the annotators as latent variables, the proposed model extends naturally to sequence labelings problems, leading to a tractable model that generalizes CRFs to multiple-annotator settings. In Section 3.2, we introduce this concept of *latent expertise models* in the context of logistic regression classifiers, and in Section 3.3, we extend it to conditional random fields.

3.2 Distinguishing good from random annotators

A key problem in learning from multiple annotators is that of distinguishing the good ones from the less good, or even random ones, in the absence of ground truth labels. This is particularly important in crowdsourcing platforms like AMT, since it allows us to reward good workers and ban, or even deny payment, to random workers. In this section, we shall formalize this idea into a probabilistic model that treats the expertise of the different annotators as latent variables. But first, let us motivate it by analyzing in more detail some of the problems that can arise when using a latent ground truth model.

3.2.1 The problem with latent ground truth models

Let us consider the popular latent ground truth model proposed by Raykar et al. (2009, 2010). Extending this model to multi-class problems leads to the following generative process:

1. For each instance $\mathbf{x}_n$
   (a) Draw true class $c_n|\mathbf{x}_n, \boldsymbol{\eta} \sim \text{Multinomial}(c_n|\text{Softmax}(\mathbf{x}_n, \boldsymbol{\eta}))$
   (b) For each annotator r
      i. Draw annotator's answer $y_n^r|c_n, \boldsymbol{\Pi}^r \sim \text{Multinomial}(y_n^r|\boldsymbol{\pi}_{c_n}^r)$

The notation $\text{Softmax}(\mathbf{x}_n, \boldsymbol{\eta})$ is used to denote a softmax function (Murphy, 2012). The softmax is the multi-dimensional generalization of the logistic sigmoid and it is defined as

$$\text{Softmax}(\mathbf{x}_n, \boldsymbol{\eta})_c = \frac{\exp(\boldsymbol{\eta}_c^{\mathrm{T}}\mathbf{x}_n)}{\sum_l \exp(\boldsymbol{\eta}_l^{\mathrm{T}}\mathbf{x}_n)}, \quad \text{for } c \in \{1, \ldots, C\} \tag{3.1}$$

where C is the number of classes. The matrix $\boldsymbol{\Pi}^r = (\boldsymbol{\pi}_1^r, \ldots, \boldsymbol{\pi}_C^r)^{\mathrm{T}}$ corresponds to the confusion matrix of the r^{th} annotator, such that the element $\pi_{c,l}^r$ corresponds to the probability that the annotator provides the label l given that the true class is c.

Following the generative process described above, the complete-data likelihood $p(\mathbf{Y}, \mathbf{c}|\mathbf{X}, \boldsymbol{\eta}, \boldsymbol{\Pi}_{1:R})$ is given by

$$p(\mathbf{Y}, \mathbf{c}|\mathbf{X}, \boldsymbol{\eta}, \boldsymbol{\Pi}_{1:R}) = \prod_{n=1}^{N} p(c_n|\mathbf{x}_n, \boldsymbol{\eta}) \prod_{r=1}^{R} p(y_n^r|c_n, \boldsymbol{\Pi}^r), \tag{3.2}$$

where $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$ and $\mathbf{Y} = \{y_n^1, \ldots, y_n^R\}_{n=1}^N$. Since the true labels $\mathbf{c}$ are not observed, we need to average over their possible values in order to obtain the marginal likelihood of the observed data $\mathcal{D} = \{\mathbf{X}, \mathbf{Y}\}$, yielding

$$p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\eta}, \boldsymbol{\Pi}_{1:R}) = \prod_{n=1}^{N} \sum_{c_n} p(c_n|\mathbf{x}_n, \boldsymbol{\eta}) \prod_{r=1}^{R} p(y_n^r|c_n, \boldsymbol{\Pi}^r). \tag{3.3}$$

Even if we consider an approach like EM for inferring the ground truth labels $\mathbf{c}$ and to estimate the model parameters $\boldsymbol{\eta}$ and $\boldsymbol{\Pi}_{1:R}$, we would still be stuck with a marginalization over the output space by averaging over all possible values for the latent true labels c_n. Although this marginalization is not problematic for classification problems where the number of classes, C, is small, for other types of problems like sequence labeling tasks (or any task with structured outputs), marginalizing over the output space is intractable in general (Sutton, 2012). If we consider, for example, the tasks of part-of-speech tagging or named entity recognition, which are usually handled as a sequence labeling problems, it is easy to see that the number of possible label sequences grows exponentially with the length of the sentence, deeming the summation over the output space intractable.

Regarding complexity, by modeling annotator expertise with a full confusion matrix $\boldsymbol{\Pi}^r$, the generative process described is able to model class-specific biases that annotators might have. However, this comes at a potential cost. Since the matrix $\boldsymbol{\Pi}^r$ comprises $C \times C$ parameters, and since in practice, on crowdsourcing platforms like AMT, the annotators frequently label a rather small set of instances, having a model with so many parameters for the reliability of the annotators can lead to overfitting. Hence, in situations where annotator biases are unlikely to occur, having a simpler model with less parameters for the annotator's expertise can be preferable.

3.2.2 Latent expertise models

Let us now propose a different class of models for learning from multiple annotators, which we refer to as *latent expertise models*. The idea is to encode the information regarding whether or not the r^{th} annotator is labeling the n^{th} instance correctly using a latent binary variable z_n^r. Hence, $z_n^r \sim \text{Bernoulli}(\phi^r)$, where the parameter ϕ^r corresponds to the accuracy of the r^{th} annotator. The expected value of this Bernoulli random variable $\mathbb{E}[z_n^r] = p(z_n^r = 1|\phi^r)$ can then be interpreted as the probability of an annotator providing a correct label or, in other words, as an indicator of how reliable an annotator is. This is a key difference between this model and latent ground truth models such as the ones by Dawid and Skene (1979) and Raykar et al. (2010). While, the latter approaches model the annotators' expertise using a full confusion matrix, the proposed model keeps a single accuracy parameter ϕ^r per annotator.

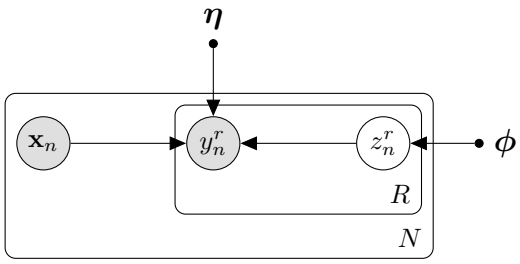


Figure 3.4: Graphical model of the proposed latent expertise model.

For the sake of simplicity, let us assume that unreliable annotators provide labels at random with uniform probability and that good annotators provide labels according to a multi-class logistic regression model on the inputs $\mathbf{x}_n$. The generative process can then be defined as follows:

1. For each instance $\mathbf{x}_n$
   (a) For each annotator r
      i. Draw reliability indicator $z_n^r|\phi^r \sim \text{Bernoulli}(z_n^r|\phi^r)$
      ii. If $z_n^r = 1$
         Draw answer $y_n^r|\mathbf{x}_n, \boldsymbol{\eta} \sim \text{Multinomial}(y_n^r|\text{Softmax}(\mathbf{x}_n, \boldsymbol{\eta}))$
      iii. If $z_n^r = 0$
         Draw answer $y_n^r \sim \text{Uniform}(y_n^r)$

Following the generative process, we can define the conditional probability distribution, $p(y_n^r|\mathbf{x}_n, z_n^r, \boldsymbol{\eta})$, as

$$p(y_n^r|\mathbf{x}_n, z_n^r, \boldsymbol{\eta}) \triangleq \left(\frac{\exp(\boldsymbol{\eta}_{y_n^r}^{\mathrm{T}}\mathbf{x}_n)}{\sum_l \exp(\boldsymbol{\eta}_l^{\mathrm{T}}\mathbf{x}_n)}\right)^{z_n^r}\left(\frac{1}{C}\right)^{1-z_n^r}, \tag{3.4}$$

where C is the number of classes. This can be verified by assigning values to z_n^r. If we set $z_n^r = 1$, the likelihood of a multi-class logistic regression is recovered and, conversely, if we set $z_n^r = 0$, we get the uniform model, as desired.

Figure 3.4 shows the probabilistic graphical model corresponding to the proposed generative process. According to the graphical model, the complete-data likelihood factorizes as

$$p(\mathbf{Y}, \mathbf{Z}|\mathbf{X}, \boldsymbol{\phi}, \boldsymbol{\eta}) = \prod_{n=1}^{N}\prod_{r=1}^{R} p(z_n^r|\phi^r)\, p(y_n^r|\mathbf{x}_n, z_n^r, \boldsymbol{\eta}), \tag{3.5}$$

where $\mathbf{Z} = \{\mathbf{z}^r\}_{r=1}^R$ with $\mathbf{z}^r = \{z_n^r\}_{n=1}^N$, and $\boldsymbol{\phi} = \{\phi^1, \ldots, \phi^R\}$. Notice that, similarly to previous works (e.g. Dawid and Skene (1979); Raykar et al. (2010); Yan et al. (2010)), we are assuming that the annotators make their decisions independently of each other. This is in general a reasonable assumption.

Since the latent indicator variables z_n^r are not observed, we need to average over their possible values. The marginal likelihood of the data is then given by

$$p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\phi}, \boldsymbol{\eta}) = \prod_{n=1}^{N}\prod_{r=1}^{R}\sum_{z_n^r \in \{0,1\}} p(z_n^r|\phi^r)\, p(y_n^r|\mathbf{x}_n, z_n^r, \boldsymbol{\eta}). \tag{3.6}$$

Our goal is to estimate the parameters $\{\boldsymbol{\phi}, \boldsymbol{\eta}\}$.

3.2.3 Estimation

As with other latent variable models, we rely on expectation-maximization (EM) (Dempster et al., 1977) to infer the posterior distribution of the latent variables z_n^r and to estimate the parameters $\boldsymbol{\phi}$ and $\boldsymbol{\eta}$.

If we observed the complete dataset $\{\mathbf{X}, \mathbf{Y}, \mathbf{Z}\}$, the log likelihood would simply be given by $\log p(\mathbf{Y}, \mathbf{Z}|\mathbf{X}, \boldsymbol{\phi}, \boldsymbol{\eta})$. Since we only have access to the "incomplete" dataset $\{\mathbf{X}, \mathbf{Y}\}$, our state of the knowledge about the values of $\mathbf{Z}$ (the reliabilities of the annotators) can be given by the posterior distribution $p(\mathbf{Z}|\mathbf{X}, \mathbf{Y}, \boldsymbol{\phi}, \boldsymbol{\eta})$. Let $q(\mathbf{Z})$ denote an estimate of the posterior distribution $p(\mathbf{Z}|\mathbf{X}, \mathbf{Y}, \boldsymbol{\phi}, \boldsymbol{\eta})$. In EM, instead of considering the complete data log likelihood, $\log p(\mathbf{Y}, \mathbf{Z}|\mathbf{X}, \boldsymbol{\phi}, \boldsymbol{\eta})$, we consider its expected value under our current estimate of the posterior distribution of the latent variables $q(\mathbf{Z})$ given by

$$\begin{aligned}\mathbb{E}_{q(\mathbf{Z})}[\log p(\mathbf{Y}, \mathbf{Z}|\mathbf{X}, \boldsymbol{\phi}, \boldsymbol{\eta})] &= \sum_{\mathbf{Z}} q(\mathbf{Z}) \log p(\mathbf{Y}, \mathbf{Z}|\mathbf{X}, \boldsymbol{\phi}, \boldsymbol{\eta}) \\ &= \sum_{n=1}^{N} \sum_{r=1}^{R} \sum_{z_n^r \in \{0,1\}} q(z_n^r) \log \big(p(z_n^r|\phi^r)\, p(y_n^r|\mathbf{x}_n, z_n^r, \boldsymbol{\eta}) \big).\end{aligned} \tag{3.7}$$

Given a current estimate of the parameters $\{\boldsymbol{\phi}, \boldsymbol{\eta}\}$, the posterior distribution of the individual latent variables $p(z_n^r|\mathbf{x}_n, y_n^r, \boldsymbol{\phi}, \boldsymbol{\eta})$, which we abbreviate as $q(z_n^r)$, can be estimated using the Bayes theorem, yielding

$$\begin{aligned}q(z_n^r = 1) &= \frac{p(z_n^r = 1|\phi^r)\, p(y_n^r|\mathbf{x}_n, z_n^r = 1, \boldsymbol{\eta})}{p(z_n^r = 1|\phi^r)\, p(y_n^r|\mathbf{x}_n, z_n^r = 1, \boldsymbol{\eta}) + p(z_n^r = 0|\phi^r)\, p(y_n^r|\mathbf{x}_n, z_n^r = 0, \boldsymbol{\eta})} \\ &= \frac{\phi^r \exp(\boldsymbol{\eta}_{y_n^r}^{\mathrm{T}} \mathbf{x}_n) / \sum_l \exp(\boldsymbol{\eta}_l^{\mathrm{T}} \mathbf{x}_n)}{\phi^r \exp(\boldsymbol{\eta}_{y_n^r}^{\mathrm{T}} \mathbf{x}_n) / \sum_l \exp(\boldsymbol{\eta}_l^{\mathrm{T}} \mathbf{x}_n) + (1 - \phi^r)(1/C)},\end{aligned} \tag{3.8}$$

where we made use of (3.4).

The expected value of the complete data log likelihood then becomes

$$\begin{aligned}\mathbb{E}_{q(\mathbf{Z})}\Big[\log p(\mathbf{Y}, \mathbf{Z}|\mathbf{X}, \boldsymbol{\phi}, \boldsymbol{\eta}) \Big] &= \sum_{n=1}^{N} \sum_{r=1}^{R} q(z_n^r = 1) \log \left(\phi^r \frac{\exp(\boldsymbol{\eta}_{y_n^r}^{\mathrm{T}} \mathbf{x}_n)}{\sum_l \exp(\boldsymbol{\eta}_l^{\mathrm{T}} \mathbf{x}_n)} \right) \\ &\quad + \sum_{n=1}^{N} \sum_{r=1}^{R} \Big(1 - q(z_n^r = 1) \Big) \log \left((1 - \phi^r) \frac{1}{C} \right).\end{aligned} \tag{3.9}$$

In the M-step of the EM algorithm, we use this expectation to estimate the model parameters.

The EM algorithm can then be summarized as follows:

E-step Compute the posterior distribution of the latent variables z_n^r using (3.8).

M-step Estimate the new model parameters $\boldsymbol{\eta}^{\text{new}}$ and $\boldsymbol{\phi}^{\text{new}}$ given by

$$\boldsymbol{\eta}^{\text{new}} = \arg\max_{\boldsymbol{\eta}} \sum_{n=1}^{N} \sum_{r=1}^{R} q(z_n^r = 1) \log \left(\frac{\exp(\boldsymbol{\eta}_{y_n^r}^{\mathrm{T}} \mathbf{x}_n)}{\sum_l \exp(\boldsymbol{\eta}_l^{\mathrm{T}} \mathbf{x}_n)} \right) \tag{3.10}$$

$$(\phi^r)^{\text{new}} = \frac{1}{N} \sum_{n=1}^{N} \mathbb{I}(y_n^r = \hat{c}_n), \tag{3.11}$$

where the estimated ground truth labels $\hat{c}_n$ are given by $\hat{c}_n = \arg\max_c (\boldsymbol{\eta}_c^{\mathrm{T}} \mathbf{x}_n)$.

Since taking derivatives of (3.9) w.r.t. $\boldsymbol{\eta}$ and setting them to zero does not yield a closed-form solution, we use a numerical optimization procedure, namely limited-memory BFGS (Liu and Nocedal, 1989), to find a maximum. The gradient is given by

$$\nabla_{\boldsymbol{\eta}_c} = \sum_{n=1}^{N} \sum_{r=1}^{R} q(z_n^r = 1) \left(\mathbb{I}(y_n^r = c) - \frac{\exp(\boldsymbol{\eta}_c^{\mathrm{T}} \mathbf{x}_n)}{\sum_l \exp(\boldsymbol{\eta}_l^{\mathrm{T}} \mathbf{x}_n)} \right) \mathbf{x}_n. \tag{3.12}$$

Notice that this is very similar to the standard training of a multi-class logistic regression model. However, in this case, the contributions to the log likelihood of the labels provided by the different annotators are being "weighted" by their reliabilities, or in other words, by how likely they are to be correct. This makes our proposed approach quite easy to implement in practice.

3.3 Sequence labeling with multiple annotators

Despite the variety of approaches presented for learning from multiple annotators under different paradigms, the problem of sequence labeling using multiple-annotator data was left practically untouched, with the only relevant work being done by Dredze et al. (2009). In this work the authors propose a method for learning structured predictors, namely conditional random fields (CRFs), from instances with multiple labels in the presence of noise. This is achieved by modifying the CRF objective function used for training through the inclusion of a per-label prior, thereby restricting the model from straying too far from the provided priors. The per-label priors are then re-estimated by making use of their likelihoods under the whole dataset. In this way, the model is capable of using knowledge from other parts of the dataset to prefer certain labels over others. By iterating between the computation of the expected values of the label priors and the estimation of the model parameters in an EM-like style, the model is expected to give preference to the less noisy labels. Hence, we can view this process as self-training, i.e. a process whereby the model is trained iteratively on its own output. Although this approach makes the model computationally tractable, their experimental results indicate that this method only improves performance in scenarios where there is a small amount of training data (low quantity) and when the labels are noisy (low quality).

It is important to stress that, contrarily to the model proposed in this section, the model by Dredze et al. (2009) is a multi-label model, and not a multi-annotator model, in the sense that the knowledge about who provided the multiple label sequences is completely discarded. The obvious solution for including this knowledge would be to use a latent ground truth model similar to the one proposed by Raykar et al. (2009, 2010), thus extending that work to sequence labeling tasks. However, as discussed in Section 3.2.1, treating the ground truth label sequences as latent variables and using an EM algorithm to estimate the model parameters would be problematic, since the number of possible label sequences grows exponentially with the length of the sequence, making the marginalization over the latent variables intractable. In contrast, in this section we extend the idea of latent expertise models that we developed in Section 3.2.2 for logistic regression to CRFs, thus leading to a tractable solution.

3.3.1 Conditional random fields

Let $\mathbf{x}_n = \{x_{n,t}\}_{t=1}^T$ be a sequence of discrete input variables of length T, and let $\mathbf{c}_n = \{c_{n,t}\}_{t=1}^T$ be a sequence of labels, corresponding to labels of each element in the input sequence $\mathbf{x}_n$. If we consider, for example, the problem of POS tagging, the variable $x_{n,t}$ can be regarded as the t^{th} word in the n^{th} sentence and $c_{n,t}$ as its corresponding part-of-speech (e.g. noun, verb, adjective, etc.). If for a dataset of N input sequences $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$ we knew the corresponding correct label sequences $\mathbf{C} = \{\mathbf{c}_n\}_{n=1}^N$, we could model the probabilities of the label sequences $\mathbf{C}$ given the input sequences $\mathbf{X}$ using a linear-chain CRF (Lafferty et al., 2001). Although we shall focus on linear-chain CRFs, such as those commonly used for NER and POS tagging, it is important to point out that the proposed approach is equally applicable to general CRFs.

In a linear-chain CRF the conditional probability of a sequence of labels $\mathbf{c}$ given a sequence of observations $\mathbf{x}$ is given by

$$p_{\text{crf}}(\mathbf{c}|\mathbf{x}, \boldsymbol{\eta}) = \frac{1}{Z(\mathbf{x}, \boldsymbol{\eta})} \prod_{t=1}^{T} \exp\left(\sum_{k=1}^{K} \eta_k \, f_k(c_{t-1}, c_t, \mathbf{x}, t) \right), \tag{3.13}$$

where T is the length of the sequence, K is an arbitrary number of features, $f_k(c_{t-1}, c_t, \mathbf{x}, t)$ is a feature function (often binary-valued, but that can also be real-valued), η_k is a learned weight associated with feature f_k, and $Z(\mathbf{x}, \boldsymbol{\eta})$ is an input-dependent normalization function that makes the probability of all label sequences sum to one, i.e.

$$Z(\mathbf{x}, \boldsymbol{\eta}) = \sum_{\mathbf{c}} \prod_{t=1}^{T} \exp\left(\sum_{k=1}^{K} \eta_k \, f_k(c_{t-1}, c_t, \mathbf{x}, t) \right). \tag{3.14}$$

The feature functions can capture any aspect of the state transitions $c_{t-1} \rightarrow c_t$ and of the whole input sequence $\mathbf{x}$, which in fact, can be used to understand the relationship between labels and the characteristics of the whole input sequence $\mathbf{x}$ at a given moment t.

According to the model defined in (3.13), the most probable labeling sequence for an input sequence $\mathbf{x}$ is given by $\mathbf{c}_* = \arg\max_{\mathbf{c}} p_{\text{crf}}(\mathbf{c}|\mathbf{x}, \boldsymbol{\eta})$, which can be efficiently determined through dynamic programming using the Viterbi algorithm (see Sutton (2012) for the details on how to perform exact inference on linear-chain CRFs).

The parameters $\boldsymbol{\eta}$ of a CRF model are typically estimated from an i.i.d. dataset by a maximum-a-posteriori (MAP) procedure. Assuming a zero-mean Gaussian prior with σ^2 variance on each individual η_k, such that $\eta_k \sim \mathcal{N}(\eta_k|0, \sigma^2)$, the posterior on $\boldsymbol{\eta}$ is proportional to

$$p(\boldsymbol{\eta}|\mathbf{X}, \mathbf{C}, \sigma^2) \propto \underbrace{\left(\prod_{n=1}^{N} p_{\text{crf}}(\mathbf{c}_n|\mathbf{x}_n, \boldsymbol{\eta}) \right)}_{\text{likelihood}} \underbrace{\prod_{k=1}^{K} \mathcal{N}(\eta_k|0, \sigma^2)}_{\text{prior}}. \tag{3.15}$$

We can find a MAP estimate of the parameters by maximizing the logarithm of (3.15) w.r.t. $\boldsymbol{\eta}$

$$\boldsymbol{\eta}_{\text{MAP}} = \arg\max_{\boldsymbol{\eta}} \sum_{n=1}^{N} \sum_{t=1}^{T} \sum_{k=1}^{K} \eta_k \, f_k(c_{n,t-1}, c_{n,t}, \mathbf{x}_n, t) - \sum_{n=1}^{N} \log Z(\mathbf{x}_n, \boldsymbol{\eta}) - \sum_{k=1}^{K} \frac{\eta_k^2}{2\sigma^2},$$

where we made use of (3.13) and the definition of a Gaussian in (A.12). Typically, a numerical optimization procedure, such as limited-memory BFGS (Liu and Nocedal, 1989), is used to find an optimum.

3.3.2 Proposed model

Let $\mathbf{y}^r$ be a sequence of labels assigned by the r^{th} annotator to some observed input sequence $\mathbf{x}$. If we were told the actual (unobserved) sequence of true labels $\mathbf{c}$ for that same input sequence $\mathbf{x}$, we could evaluate the quality, or reliability, of the r^{th} annotator in a dataset by measuring its precision and recall. Furthermore, we could combine precision and recall in a single measure by using the traditional F1-measure, and use this combined measure to evaluate how "good" or "reliable" a given annotator is according to some ground truth. In practice any appropriate loss function can be used to evaluate the quality of the annotators. The choice of one metric over others is purely problem-specific. The F-measure is considered here due to its wide applicability in sequence labeling problems and, particularly, in the tasks used in the experiments (Section 3.4).

Since we do not know the set of actual ground truth label sequences $\mathbf{C}$ for the input sequences $\mathbf{X}$, we must find a way to estimate it using the sets of label sequences provided by the R different annotators $\{\mathbf{Y}^1, ..., \mathbf{Y}^R\}$, and learn a CRF model along the way. For that, we shall consider a slightly different approach to the one we developed in Section 3.2.2 for logistic regression. Namely, instead of considering that, when labeling the n^{th} instance, each annotator r flips a biased coin (represented through the variables z_n^r) to decide whether or not to provide the correct label, we shall consider that the annotators throw a die to determine which one of them will provide the correct label sequences. Let z be the outcome of that die draw and let $\text{F1-measure}_r(\mathbf{Y}^r, \mathbf{C})$ denote the F1-measure of the answers of the r^{th} annotator, $\mathbf{Y}^r$, evaluated against the ground truth label sequences $\mathbf{C}$. Considering the F1-measure to be a good indicator of how reliable, or how likely an annotator is to provide correct label sequences, we can assume that $z \sim \text{Multinomial}(\boldsymbol{\phi})$, i.e., z has a multinomial distribution with parameters $\boldsymbol{\phi} = (\phi^1, \ldots, \phi^R)^{\text{T}}$, where we define

$$\phi^r \triangleq \frac{\text{F1-measure}_r(\mathbf{Y}^r, \mathbf{C})}{\sum_{j=1}^{R} \text{F1-measure}_j(\mathbf{Y}^j, \mathbf{C})}, \tag{3.16}$$

thus ensuring the constraints $\phi_r \geq 0$ (since the F1-measure is always non-negative) and $\sum_r \phi_r = 1$. Depending on the value of the variable z, the annotator then decides whether or not to provide correct labels.

The expectation $\mathbb{E}[z^r] = p(z^r = 1)$ can therefore be interpreted as the probability of picking the label sequences provided by the r^{th} annotator as the correct ones (i.e. for which $\text{F1-measure}_r(\mathbf{Y}^r, \mathbf{C}) = 1$) and using those for training. An analogy for this model would be a student picking a book to learn about some subject. The student is provided by the university's library with a set of books that cover that subject but differ only in how good and pedagogical they are. The student then has to pick one of the books from which to learn about that subject. Transferring this analogy back to our multiple annotator setting, the random vector z can be viewed as picking the best annotator from which to learn from, thus enforcing competition among the annotators. The correct annotator is assumed to provide label sequences according to a CRF model, $p_{\text{crf}}(\mathbf{y}_n^r | \mathbf{x}_n, \boldsymbol{\eta})$. The others are assumed to provide incorrect labels

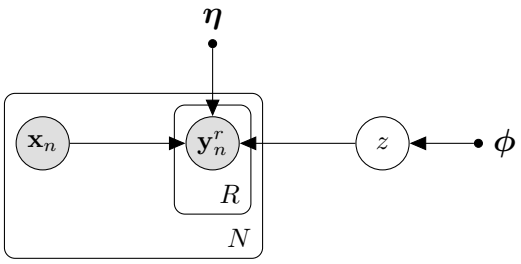


Figure 3.5: Proposed latent expertise model for sequence labeling problems.

which we assume to come from a random model $p_{\text{rand}}(\mathbf{y}_n^r)$. For the sake of simplicity, we assume the random model $p_{\text{rand}}(\mathbf{y}_n^r)$ to be uniformly distributed. Hence

$$p_{\text{rand}}(\mathbf{y}_n^r) = \prod_{t=1}^{T} \frac{1}{C}, \tag{3.17}$$

where T denotes the length of the sequence and C is the number of possible classes for a sequence element[1].

The generative process can then be summarized as follows:

1. Draw latent variable $z|\boldsymbol{\phi} \sim \text{Multinomial}(z|\boldsymbol{\phi})$
2. For each instance $\mathbf{x}_n$
   (a) For each annotator r
      i. If $z^r = 1$:
         Draw annotator's answer $\mathbf{y}_n^r|\mathbf{x}_n, \boldsymbol{\eta} \sim p_{\text{crf}}(\mathbf{y}_n^r|\mathbf{x}_n, \boldsymbol{\eta})$
      ii. If $z^r = 0$:
         Draw annotator's answer $\mathbf{y}_n^r \sim p_{\text{rand}}(\mathbf{y}_n^r)$

Figure 3.5 shows a plate representation of the proposed model.

Although it might seem too restrictive to assume that only one annotator provides the correct label sequences, it is important to note that the model can still capture the uncertainty regarding who the correct annotator should be. In alternative to this approach, one could replace the multinomial random variable z with multiple Bernoullis z_n^r (one for each annotator) as we did in Section 3.2.2 for classification problems. From a generative perspective, this would allow for multiple annotators to be correct. However, it places too much emphasis on the form of $p_{\text{rand}}(\mathbf{y}_n^r)$, since it would be crucial for deciding whether the annotator is likely to be correct. If we recall the posterior distribution of z_n^r given by Eq. 3.8, we can see that if the value of $p_{\text{rand}}(y_n^r) = p(y_n^r|\mathbf{x}_n, z_n^r = 0, \boldsymbol{\eta})$ is too large or too small, the value of the fraction will be close to constant throughout all annotators, thus making the model believe that they are all equally reliable. While this was not a problem for classification tasks, it turns out that, for sequence labeling, a poor choice of $p_{\text{rand}}(\mathbf{y}_n^r)$ leads to poor results. On the other hand, as we shall see later, by using a multinomial distribution, the probabilities $p_{\text{rand}}(\mathbf{y}_n^r)$ cancel out from the posterior

[1]Not to be confused with $\mathbf{C}$, which denotes the set of ground truth label sequences.

distribution of z, thus forcing the annotators to "compete" with each other for the best label sequences.

Following the generative process described above, we can define

$$p(\mathbf{y}_n^1, \ldots, \mathbf{y}_n^R | \mathbf{x}_n, z, \boldsymbol{\eta}) = \prod_{r=1}^{R} p(\mathbf{y}_n^r | \mathbf{x}_n, z^r, \boldsymbol{\eta}) \\ = \prod_{r=1}^{R} \Big(p_{\text{crf}}(\mathbf{y}_n^r | \mathbf{x}_n, \boldsymbol{\eta}) \Big)^{(z^r)} \Big(p_{\text{rand}}(\mathbf{y}_n^r) \Big)^{(1-z^r)}, \tag{3.18}$$

where we made use of the assumption that the annotators make their decisions independently of each other.

If we observed the complete data $\{\mathbf{X}, \mathbf{Y}^1, \ldots, \mathbf{Y}^R, z\}$, then the likelihood would be given by the following expression

$$p(\mathbf{Y}^1, \ldots, \mathbf{Y}^R, z | \mathbf{X}, \boldsymbol{\phi}, \boldsymbol{\eta}) = p(z|\boldsymbol{\phi}) \prod_{n=1}^{N} p(\mathbf{y}_n^1, ..., \mathbf{y}_n^R | \mathbf{x}_n, z, \boldsymbol{\eta}). \tag{3.19}$$

Since we do not actually observe z, we must marginalize over it by summing over all its possible values. The likelihood of our model then becomes

$$p(\mathbf{Y}^1, \ldots, \mathbf{Y}^R | \mathbf{X}, \boldsymbol{\phi}, \boldsymbol{\eta}) = \sum_{z} p(z|\boldsymbol{\phi}) \prod_{n=1}^{N} p(\mathbf{y}_n^1, ..., \mathbf{y}_n^R | \mathbf{x}_n, z, \boldsymbol{\eta}). \tag{3.20}$$

Recalling that z is multinomial variable represented using a 1-of-K coding, we can re-write the summation in the equation above as

$$p(\mathbf{Y}^1, \ldots, \mathbf{Y}^R | \mathbf{X}, \boldsymbol{\phi}, \boldsymbol{\eta}) = \sum_{r} p(z^r = 1|\boldsymbol{\phi}) \prod_{n=1}^{N} p(\mathbf{y}_n^1, ..., \mathbf{y}_n^R | \mathbf{x}_n, z^r = 1, \boldsymbol{\eta}). \tag{3.21}$$

Making use of (3.18) and the fact that $p(z^r = 1|\boldsymbol{\phi}) = \phi^r$, the likelihood can be further simplified giving

$$p(\mathbf{Y}^1, \ldots, \mathbf{Y}^R | \mathbf{X}, \boldsymbol{\phi}, \boldsymbol{\eta}) = \sum_{r=1}^{R} \phi^r \prod_{n=1}^{N} p_{\text{crf}}(\mathbf{y}_n^r | \mathbf{x}_n, \boldsymbol{\eta}) \prod_{j \neq r} p_{\text{rand}}(\mathbf{y}_n^j | \mathbf{x}_n). \tag{3.22}$$

3.3.3 Estimation

As with the latent expertise model presented in Section 3.2, we rely on the EM algorithm (Dempster et al., 1977) to compute the posterior distribution of the latent variable z and to estimate the parameters $\{\boldsymbol{\phi}, \boldsymbol{\eta}\}$ of the proposed model.

The expectation of the complete-data log likelihood, $\log p(\mathbf{Y}^1, \ldots, \mathbf{Y}^R, z | \mathbf{X}, \boldsymbol{\phi}, \boldsymbol{\eta})$, under our current estimate of the posterior distribution of the latent variables $q(z)$ is given by

$$\mathbb{E}_{q(z)}[\log p(\mathbf{Y}^1, \ldots, \mathbf{Y}^R, z | \mathbf{X}, \boldsymbol{\phi}, \boldsymbol{\eta})] = \sum_{z} q(z) \log p(\mathbf{Y}, z | \mathbf{X}, \boldsymbol{\phi}, \boldsymbol{\eta}) \\ = \sum_{z} q(z) \log \Big(p(z|\boldsymbol{\phi}) \prod_{n=1}^{N} p(\mathbf{y}_n^1, ..., \mathbf{y}_n^R | \mathbf{x}_n, z, \boldsymbol{\eta}) \Big). \tag{3.23}$$

The posterior distribution of the latent variable z can be estimated using the Bayes' theorem

$$\begin{aligned} q(z^r = 1) &= p(z^r = 1|\mathbf{Y}^1, \ldots, \mathbf{Y}^R, \mathbf{X}, \boldsymbol{\phi}, \boldsymbol{\eta}) \\ &= \frac{p(z^r = 1|\boldsymbol{\phi})\, p(\mathbf{Y}^1, ..., \mathbf{Y}^R|\mathbf{X}, z^r = 1, \boldsymbol{\eta})}{\sum_{j=1}^{R} p(z^j = 1|\boldsymbol{\phi})\, p(\mathbf{Y}^1, ..., \mathbf{Y}^R|\mathbf{X}, z^j = 1, \boldsymbol{\eta})} \\ &= \frac{\phi^r \prod_{n=1}^{N} \Big(p_{\text{crf}}(\mathbf{y}_n^r|\mathbf{x}_n, \boldsymbol{\eta}) \prod_{k\neq r} p_{\text{rand}}(\mathbf{y}_n^k|\mathbf{x}_n) \Big)}{\sum_{j=1}^{R} \phi^j \prod_{i=1}^{N} \Big(p_{\text{crf}}(\mathbf{y}_n^j|\mathbf{x}_n, \boldsymbol{\eta}) \prod_{k\neq j} p_{\text{rand}}(\mathbf{y}_n^k|\mathbf{x}_n) \Big)}. \end{aligned} \tag{3.24}$$

As long as we are assuming a uniform model for $p_{\text{rand}}(\mathbf{y}_n^r)$, we have that $p_{\text{rand}}(\mathbf{y}_n^r) = p_{\text{rand}}(\mathbf{y}_n^j), \forall r, j \in \{1, \ldots, R\}$. Hence, the expression for the posterior distribution can be further simplified, yielding

$$q(z^r = 1) = \frac{\phi^r \prod_{n=1}^{N} p_{\text{crf}}(\mathbf{y}_n^r|\mathbf{x}_n, \boldsymbol{\eta})}{\sum_{j=1}^{R} \phi^j \prod_{n=1}^{N} p_{\text{crf}}(\mathbf{y}_n^j|\mathbf{x}_n, \boldsymbol{\eta})}. \tag{3.25}$$

Making use of the same results that led to (3.22), the expected value of the log likelihood in (3.23) becomes

$$\begin{aligned} &\mathbb{E}_{q(z)}[\log p(\mathbf{Y}^1, \ldots, \mathbf{Y}^R, z|\mathbf{X}, \boldsymbol{\phi}, \boldsymbol{\eta})] \\ &= \sum_{r=1}^{R} q(z^r = 1) \Bigg(\log \phi^r + \sum_{n=1}^{N} \Big(\log p_{\text{crf}}(\mathbf{y}_n^r|\mathbf{x}_n, \boldsymbol{\eta}) + \sum_{j\neq r} \log p_{\text{rand}}(\mathbf{y}_n^j|\mathbf{x}_n) \Big) \Bigg). \end{aligned} \tag{3.26}$$

In the M-step of the EM algorithm, we estimate the new model parameters $\boldsymbol{\phi}^{new}$ and $\boldsymbol{\eta}^{new}$. As we did in (3.16) for standard linear-chain CRFs, we place a zero-mean Gaussian prior on $\boldsymbol{\eta}$ with σ^2 variance, which prevents the coefficients $\boldsymbol{\eta}$ (weights) from becoming arbitrarily large. This is a widely known form of regularlization, which is commonly referred to as ℓ_2-regularization (Ng, 2004). The strength of the regularization is controlled by the value of σ^2. The regularized log likelihood is then given by

$$\begin{aligned} \mathbb{E}_{q(z)}[\log p(\mathbf{Y}^1, \ldots, \mathbf{Y}^R, z|\mathbf{X}, \boldsymbol{\phi}, \boldsymbol{\eta})] = \sum_{r=1}^{R} q(z^r = 1) \Bigg(\log \phi^r + \sum_{n=1}^{N} \Big(\log p_{\text{crf}}(\mathbf{y}_n^r|\mathbf{x}_n, \boldsymbol{\eta}) \\ + \sum_{j\neq r} \log p_{\text{rand}}(\mathbf{y}_n^j|\mathbf{x}_n) \Big) \Bigg) + \sum_{k=1}^{K} \log \mathcal{N}(\eta_k|0, \sigma^2). \end{aligned} \tag{3.27}$$

The EM algorithm can then be summarized as follows:

E-step Compute the posterior distribution of the latent variable z using (3.25).

M-step Estimate new model parameters $\boldsymbol{\eta}^{new}$ and $\boldsymbol{\phi}^{new}$ as

$$\boldsymbol{\eta}^{new} = \arg\max_{\boldsymbol{\eta}} \sum_{n=1}^{N} \sum_{r=1}^{R} q(z^r = 1) \Big(\log p_{\text{crf}}(\mathbf{y}_n^r|\mathbf{x}_n, \boldsymbol{\eta}) \Big) - \sum_{k=1}^{K} \frac{\eta_k^2}{2\sigma^2}, \tag{3.28}$$

$$(\phi^r)^{new} = \frac{\text{F1-measure}_r(\mathbf{Y}^r, \mathbf{C})}{\sum_{j=1}^{R} \text{F1-measure}_j(\mathbf{Y}^j, \mathbf{C})}, \tag{3.29}$$

where the estimated ground truth labels are given by $\mathbf{c}_n = \arg\max_{\mathbf{c}_n} p_{\text{crf}}(\mathbf{c}_n|\mathbf{x}_n, \boldsymbol{\eta}^{new})$, which can be efficiently determined using the Viterbi algorithm[2]. In (3.28), the new CRF model parameters $\boldsymbol{\eta}^{new}$ are determined using limited-memory BFGS similarly to normal CRF training (Sutton and McCallum, 2006). However, the log likelihood function now includes a weighting factor: $q(z^r = 1)$. From this perspective, when learning from label sequences of various annotators, the proposed model is weighting the latter by how much it expects them to be right, while considering also how likely the other annotators are to be correct. If, for example, there are only two "good" annotators, they will share the responsibility in "teaching" the CRF model.

The initialization of the EM algorithm can be simply done by assigning random values to the annotators reliabilities or by estimating the ground truth label sequences $\mathbf{C}$ using majority voting. The algorithm stops when the expectation in (3.23) converges or when the changes to the annotators reliabilities fall below a given threshold.

3.4 Experiments

The proposed models, multiple-annotator logistic regression (MA-LR) and multiple-annotator conditional random fields (MA-CRF)[3], from Sections 3.2 and 3.3, respectively, were evaluated using both multiple-annotator data with simulated annotators and data manually labelled using AMT. The following sections describe these experiments.

Multiple-annotator logistic regression

The proposed MA-LR model was compared with the multi-class extension of the model proposed by Raykar et al. (2010) (see Section 3.2.1), which, as we previously discussed, is a latent ground truth model, and with two majority voting baselines:

- Soft majority voting (MVsoft): this corresponds to a multi-class logistic regression model trained with the *soft* probabilistic labels resultant from the voting process.
- Hard majority voting (MVhard): this corresponds to a multi-class logistic regression model trained with the most voted labels resultant from the voting process, i.e. the most voted class for a given instance gets "1" and the others get "0".

In all experiments the EM algorithm was initialized with majority voting.

Simulated annotators

With the purpose of comparing the presented approaches in different classification tasks, we used six popular benchmark datasets from the UCI repository[4] — a collection of databases, domain theories, and data generators that are used by the machine

[2]Note that the ground truth estimate is required to compute the F1-scores of the annotators and estimate the multinomial parameters $\boldsymbol{\phi}$.

[3]Source code for MA-LR and MA-CRF is available at: http://amilab.dei.uc.pt/fmpr/

[4]http://archive.ics.uci.edu/ml/index.html

Dataset	Num. Instances	Num. Features	Num. Classes
annealing	798	38	6
image segmentation	2310	19	7
ionosphere	351	34	2
iris	150	4	3
parkinsons	197	23	2
wine	178	13	3

Table 3.1: Details of the UCI datasets.

learning community for the empirical analysis of machine learning algorithms. Since these datasets do not have labels from multiple annotators, the latter were simulated from the ground truth using two different methods. The first method, denoted "label flips", consists in randomly flipping the label of an instance with a given uniform probability $p(\text{flip})$ in order to simulate an annotator with an average reliability of $1-p(\text{flip})$. The second method, referred to as "model noise", seeks simulating annotators that are more consistent in their opinions, and can be summarized as follows. First, a multi-class logistic regression model is trained on the original training set. Then, the resulting weights $\boldsymbol{\eta}$ are perturbed, such that the classifier consistently "fails" in a coherent fashion throughout the testset. To do so, the values of $\boldsymbol{\eta}$ are standardized, and then random "noise" is drawn from a Gaussian distribution with zero mean and σ^2 variance and added to the weights $\boldsymbol{\eta}$. These weights are then "unstandardized" (by reversing the standardization process previously used), and the modified multi-class logistic regression model is re-applied to the training set in order to make predictions that simulate the answers of an annotator. The quality of this annotator will vary depending on the value of σ^2 used. This processed is then repeated R times in order to simulate the answers of R independent annotators.

Since in practice each annotator only labels a small subset of all the instances in the dataset, we introduce another parameter in this annotator simulation process: the probability $p(\text{label})$ of an annotator labeling an instance.

Table 4.1 describes the UCI datasets used in these experiments. Special care was taken in choosing datasets that correspond to real data and that were among the most popular ones in the repository and, consequently, among the machine learning community. Datasets that were overly unbalanced, i.e. with too many instances of some classes and very few instances of others, were avoided. Other than that, the selection process was random, which resulted in a rather heterogeneous collection of datasets: with different sizes, number of features and number of classes.

Figures 3.6 and 3.7 show the results obtained using 5 simulated annotators with different reliabilities using the simulation methods described above: "label flips" and "model noise", respectively. All the experiments use 10-fold cross-validation. Due to the stochastic nature of the simulation process of the annotators, each experiment was repeated 30 times and the average results were collected. The plots on the left show the root mean squared error (RMSE) between the estimated annotators accuracies and their actual accuracies evaluated against the ground truth. The plots on the center and on the right show, respectively, the trainset and testset accuracies. Note that, here, unlike in "typical" supervised learning tasks, trainset accuracy is quite important since it indicates how well the models are estimating the unobserved ground truth labels from the opinions of the multiple annotators.

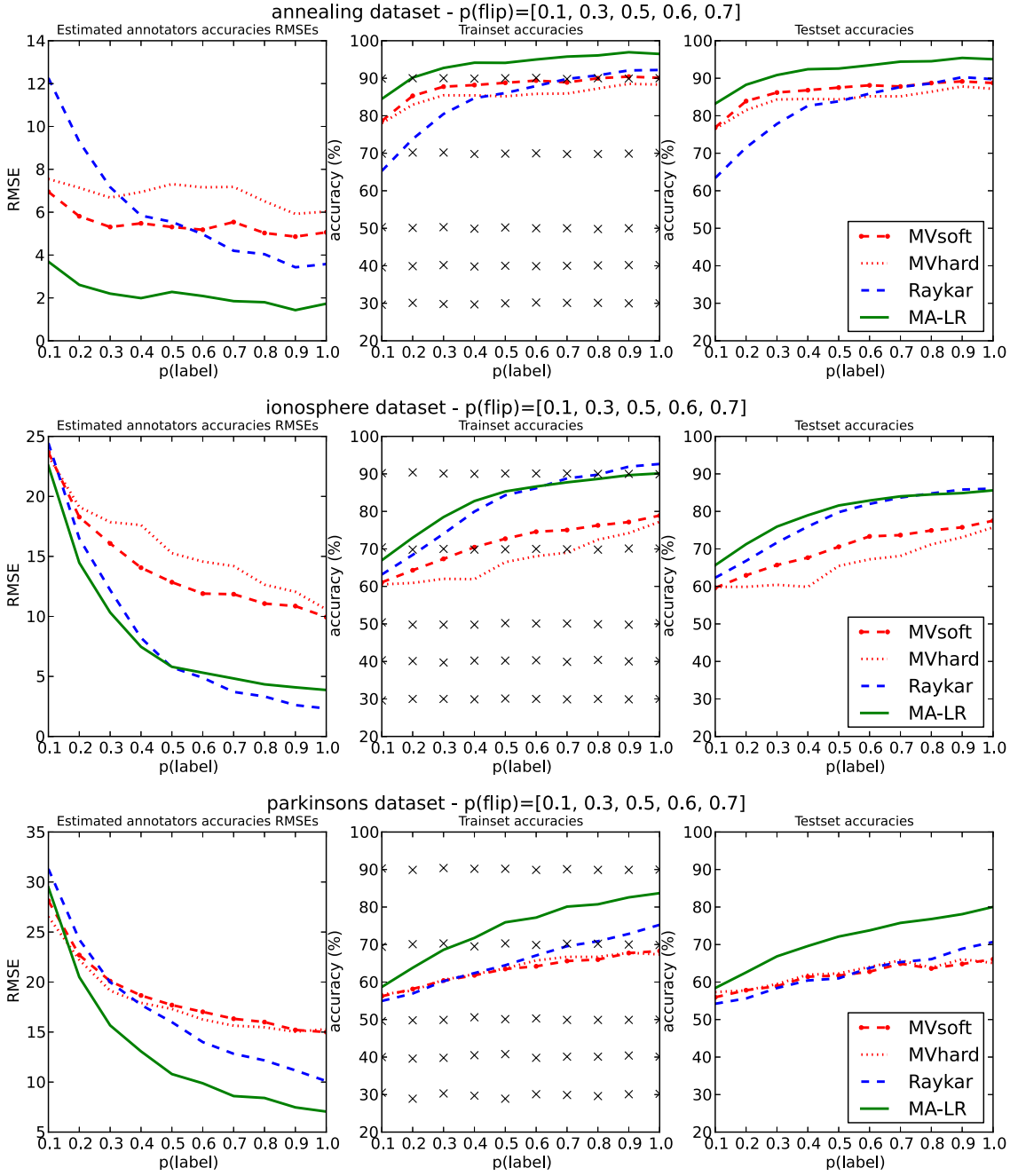


Figure 3.6: Results for the annealing, ionosphere and parkinsons datasets using the "label flips" method for simulating annotators. The "x" marks indicate the average true accuracies of the simulated annotators.

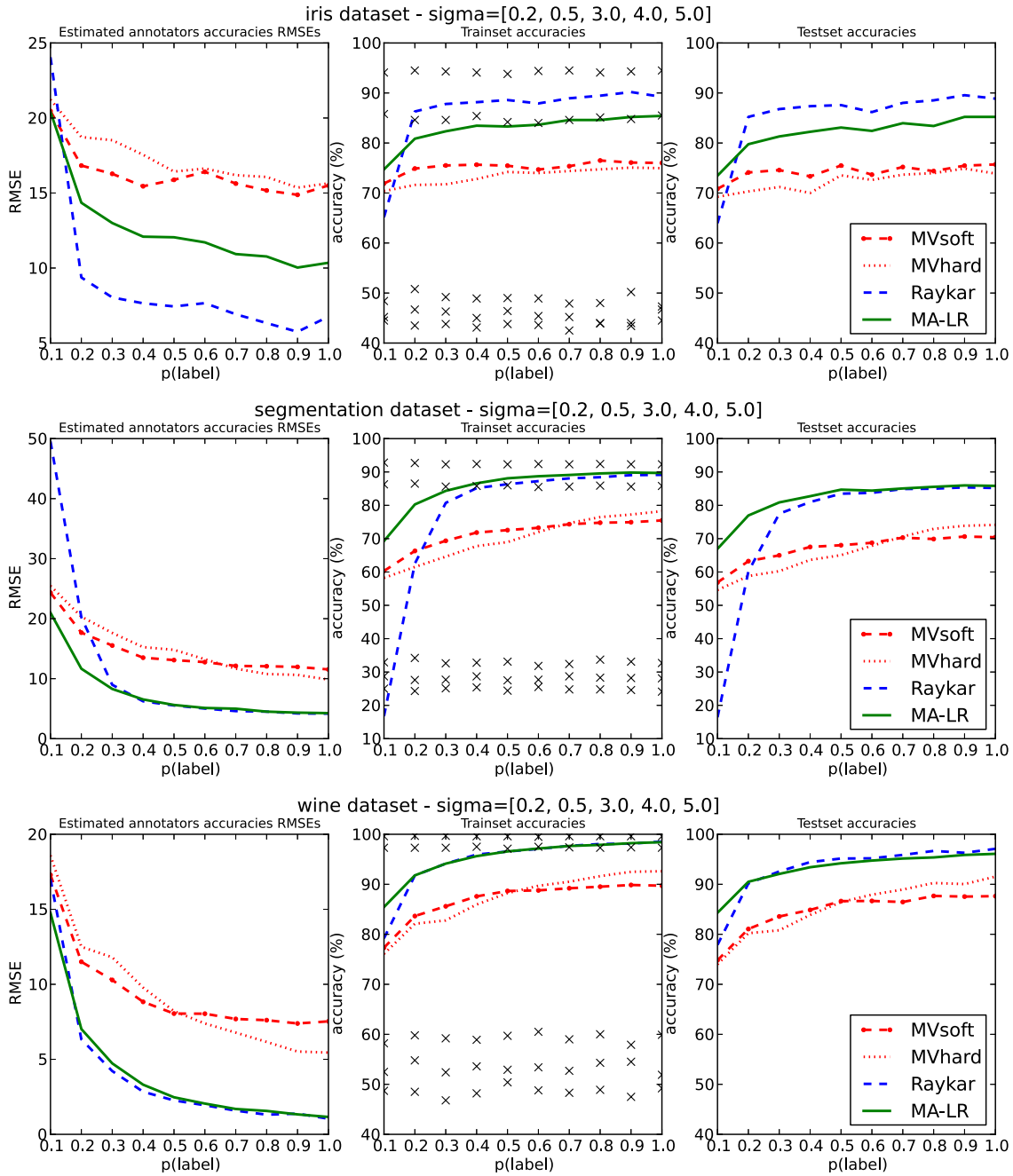


Figure 3.7: Results for the iris, segmentation and wine datasets using the "model noise" method for simulating annotators. The "x" marks indicate the average true accuracies of the simulated annotators.

From a general perspective on the results of Figures 3.6 and 3.7, we can conclude that both methods for learning from multiple annotators (MA-LR and Raykar) tend to outperform the majority voting baselines under most conditions. Not surprisingly, as the value of p(label), and consequently the average number of instances labeled by each annotator, decreases, both the trainset and testset accuracies of all the approaches decrease or stay roughly the same. As expected, a higher trainset accuracy usually translates in a higher testset accuracy and a better approximation of the annotators accuracies, i.e. lower root mean squared error (RMSE), since the approximation of the ground truth is also better.

A more careful analysis of the results reveals that, contrarily to the model by Raykar et al. (2010), the proposed model (MA-LR) is less prone to overfitting when the number of instances labeled by each annotator decreases. This is a direct consequence of the number of parameters used to model the annotators expertise. While the model by Raykar et al. (2010) uses a full $C \times C$ confusion matrix for each annotator, making a total of RC^2 parameters, the proposed model only uses R parameters. However, it is important to note that there is a tradeoff here, since the model by Raykar et al. can capture certain biases in the annotators answers, which is not possible with the MA-LR model.

Amazon mechanical turk

In order to assess the performance of MA-LR in learning from the labels of multiple non-expert human annotators and compare it with the other approaches, two experiments were conducted using AMT: sentiment polarity and music genre classification[5].

The sentiment polarity experiment was based on the sentiment analysis dataset introduced by Pang and Lee (2005), which corresponds to a collection of more than ten thousand sentences extracted from the movie reviews website RottenTomatoes[6]. These are labeled as positive or negative depending on whether they were marked as "fresh" or "rotten" respectively. From this collection, a random subset of 5000 sentences were selected and published on AMT for annotation. Given the sentences, the workers were asked to provide the sentiment polarity (positive or negative). The remaining 5428 sentences were kept for evaluation.

For the music genre classification experiment, the audio dataset introduced by Tzanetakis and Cook (2002) was used. This dataset consists of a thousand samples of songs with 30 seconds of length and divided among 10 different music genres: classical, country, disco, hiphop, jazz, rock, blues, reggae, pop and metal. Each of the genres has 100 representative samples. A random 70/30 train/test split was performed on the dataset, and the 700 training samples were published on AMT for classification. In this case, the workers were required to listen to a 30-second audio clip and classify it as one of the 10 genres enumerated above.

Table 3.2 shows some statistics about the answers of the AMT workers for both datasets. Figure 3.8 further explores the distributions of the number of answers provided by each annotator and their accuracies for the sentiment polarity and music genre datasets. The figure reveals a highly skewed distribution of number of answers per worker, which support our intuition that on this kind of crowdsourcing

[5]Datasets are available at: http://amilab.dei.uc.pt/fmpr/ma-lr/

[6]http://www.rottentomatoes.com/

	Sentiment polarity	Music genre
Number of answers collected	27747	2946
Number of workers	203	44
Avg. answers per worker (± std)	136.68 ± 345.37	66.93 ± 104.41
Min. answers per worker	5	2
Max. answers per worker	3993	368
Avg. worker accuracy (± std)	77.12 ± 17.10%	73.28 ± 24.16%
Min. worker accuracy	20%	6.8%
Max. worker accuracy	100%	100%

Table 3.2: Statistics of the answers of the AMT workers for the two experiments performed. Note that the worker accuracies correspond to trainset accuracies.

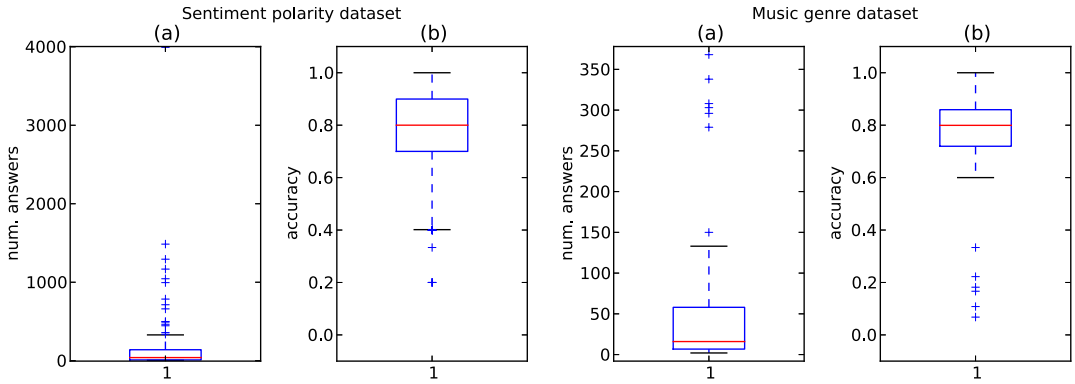


Figure 3.8: Boxplots of the number of answers (a) and the accuracies (b) of the AMT workers for the sentiment polarity (left) and music genre (right) datasets.

platforms each worker tends to only provide a small number of answers, with only a couple of workers performing high quantities of labelings.

Standard preprocessing and feature extraction techniques were performed on both experiments. In the case of the sentiment polarity dataset, the stop-words were removed and the remaining words were reduced to their root by applying a stemmer. This resulted in a vocabulary with size 8919, which still makes a bag-of-words representation computationally expensive. Hence, latent semantic analysis (LSA) was used to further reduce the dimensionally of the dataset to 1200 features.

Regarding the music genre dataset, we used Marsyas[7] — a standard music information retrieval tool — to extract a collection of commonly used features in this kind of tasks (Tzanetakis and Cook, 2002). These include means and variances of timbral features, time-domain zero-crossings, spectral centroid, rolloff, flux and mel-frequency cepstral coefficients (MFCC) over a texture window of 1 second. A total of 124 features were extracted. The details on these features fall out of the scope of this thesis. The interested reader is redirected to the appropriate literature (e.g. Aucouturier and Pachet (2003); Tzanetakis and Cook (2002)).

Table 3.3 presents the results obtained by the different methods on the sentiment polarity and music genre datasets. As expected, the results indicate that both annotator-aware methods are clearly superior when compared to the majority voting

[7] http://marsyasweb.appspot.com

	Sentiment polarity		Music genre	
Method	Train acc.	Test acc.	Train acc.	Test acc.
MVsoft	80.70%	71.65%	67.43%	60.33%
MVhard	79.68%	70.27%	67.71%	59.00%
Raykar	49.91%	48.67%	9.14%	12.00%
Raykar (w/prior)	84.92%	70.78%	71.86%	63.00%
MA-LR	**85.40%**	**72.40%**	**72.00%**	**64.00%**

Table 3.3: Trainset and testset accuracies for the different approaches on the datasets obtained from AMT.

baselines. Also, notice that due to the fact that some annotators only label a very small portion of instances, the "standard" model by Raykar et al. (2010) performs very poorly (as bad as a random classifier) due to overfitting. In order to overcome this, a prior had to be placed on the probability distribution that controls the quality of the annotators. In the case of the sentiment polarity task, a Beta$(1, 1)$ prior was used, and for the music genre task we applied a symmetric Dirichlet$(\mathbf{1}_C)$, where $\mathbf{1}_C$ denotes a vector of 1's with length C. Despite the use of a prior, the model by Raykar et al. (2010) still performs worse than the proposed MA-LR model, which takes advantage of its single quality-parameter per annotator to produce better estimates of the annotators' reliabilities. These results are coherent with our findings with the simulated annotators, which highlights the quality of the proposed model.

Multiple-annotator conditional random fields

The proposed multiple-annotator conditional random fields (MA-CRF) model was evaluated in the field of natural language processing (NLP) for the particular tasks of named entity recognition (NER) and noun phrase (NP) chunking. NER refers to the information retrieval subtask of identifying and classifying atomic elements in text into predefined categories such as the names of persons, organizations, locations and others, while NP chunking consists of recognizing chunks of sentences that correspond to noun phrases. Because of their many applications these tasks are considered very important in the field of NLP and other related areas.

As in the previous section, we perform experiments using two types of annotators: simulated annotators and real annotators from AMT. In both cases, the label sequences are represented using the traditional BIO scheme as introduced by Ramshaw and Marcus (1995), which distinguishes between the beginning of a segment (B), inside of a segment (I) or outside (O). For example, using this scheme, the NER tags for the sentence "Afonso Henriques was the first king of Portugal" will be the following: "Afonso/B-PERSON Henriques/I-PERSON was/O the/O first/O king/O of/O Portugal/B-LOCATION".

The proposed approach is compared with four baselines:

- MVseq: majority voting at sequence level (i.e., the label sequence with more votes wins);
- MVtoken: majority voting at token level (i.e., the BIO label with more votes for a given token wins);

- MVseg: this corresponds to a two-step majority voting performed over the BIO labels of the tokens. First, a majority voting is used for the segmentation process (i.e. to decide whether the token should be considered as part of a segment - a named entity for example), then a second majority voting is used to decide the labels of the segments identified (e.g. what type of named entity it is);
- CRF-CONC: a CRF using all the data instances from all annotators concatenated for training.

The proposed model is also compared with the two variants of multi-label model proposed in (Dredze et al., 2009): MultiCRF and MultiMA-CRFX. The latter differs from the former by training the CRF on the most likely (maximum) label instead of training on the (fuzzy) probabilistic labels (kindly see (Dredze et al., 2009) for the details). As an upper-bound, we also show the results of a CRF trained on ground truth (gold) data. We refer to this as "CRF-GOLD".

For all the experiments a simple set of features that is common in NLP tasks was used, namely word identity features, capitalization patterns, numeric patterns, other morphologic features (e.g. prefixes and suffixes), part-of-speech tags, bi-gram and tri-gram features and window features (window size = 3). In MA-CRF, the EM algorithm was initialized with token-level majority voting (MVtoken). The MultiCRF model was initialized with uniform label priors.

Simulated annotators

There are a few publicly available "golden" datasets for NER such as the 2003 CONLL English NER task dataset (Sang and Meulder, 2003), which is a common benchmark for sequence labeling tasks in the NLP community. Using this dataset, we obtained a trainset and a testset of 14987 and 3466 sentences respectively.

Since the 2003 CONLL shared NER dataset does not contain labels from multiple annotators, these were simulated for different reliabilities using the methods described in Section 3.4 in the context of logistic regression: "model noise" and "label flips".

Using the "model noise" method, we simulated 5 artificial annotators with $\sigma^2 = \{0.005, 0.05, 0.05, 0.1, 0.1\}$. This choice of values intends to reproduce a scenario where there is a "good", two "average" and two "bad" annotators. The proposed approach (MA-CRF) and the four baselines were then evaluated against the testset. This process was repeated 30 times and the average results are presented in Table 3.4. The results indicate that MA-CRF outperforms the four proposed baselines in both the trainset and testset. In order to assess the statistical significance of this result, after a Kolmogorov-Smirnov test verified the normality of the distributions, a paired t-test was used to compare the mean F1-score of MA-CRF in the testset against the MVseq, MVtoken, MVseg and CRF-CONC baselines. The obtained p-values were 4×10^{-25}, 7×10^{-10}, 4×2^{-8} and 1×10^{-14} respectively, which indicates that the differences are all highly significant.

Regarding the MultiCRF model, we can see that, at best, it performs almost as good as MVtoken. Not surprisingly, the "MAX" version of MultiCRF performs better than the standard version. This behavior is expected since the "hard" labels obtained from majority voting also perform better than the "soft" label effect obtained in CRF-CONC. Nonetheless, neither version of MultiCRF performs as well

	Trainset			Testset		
Method	Prec.	Recall	F1	Prec.	Recall	F1
MVseq	24.1%	50.5%	32.6 ± 2.0%	47.3%	30.9%	37.3 ± 3.1%
MVtoken	56.0%	69.1%	61.8 ± 4.1%	62.4%	62.3%	62.3 ± 3.4%
MVseg	52.5%	65.0%	58.0 ± 6.9%	60.6%	57.1%	58.7 ± 7.1%
CRF-CONC	47.9%	49.6%	48.4 ± 8.8%	47.8%	47.1%	47.1 ± 8.1%
MultiCRF	39.8%	22.6%	28.7 ± 3.8%	40.0%	15.4%	22.1 ± 5.0%
MultiMA-CRFX	55.0%	66.7%	60.2 ± 4.1%	63.2%	58.4%	60.5 ± 3.6%
MA-CRF	**72.9%**	**81.7%**	**77.0 ± 3.9%**	**72.5%**	**67.7%**	**70.1 ± 2.5%**
CRF-GOLD	99.7%	99.9%	99.8%	86.2%	87.8%	87.0%

Table 3.4: Results for the CONLL NER task with 5 simulated annotators (with $\sigma^2 = [0.005, 0.05, 0.05, 0.1, 0.1]$) with repeated labeling.

	Trainset			Testset		
Method	Precision	Recall	F1	Precision	Recall	F1
CRF-CONC	52.1%	56.5%	54.0 ± 7.3%	53.9%	51.7%	52.6 ± 6.4%
MA-CRF	**63.8%**	**71.1%**	**67.2 ± 1.7%**	**65.7%**	**62.7%**	**64.2 ± 1.6%**
CRF-GOLD	99.7%	99.9%	99.8%	86.2%	87.8%	87.0%

Table 3.5: Results for the NER task with 5 simulated annotators (with $\sigma^2 = [0.005, 0.05, 0.05, 0.1, 0.1]$) without repeated labeling.

as MA-CRF (testset p-values are 1×10^{-26} and 1×10^{-11} for the MultiCRF and MultiMA-CRFX respectively).

In order to empirically show that the proposed approach does not rely on repeated labeling, i.e. multiple annotators labeling the same data instances, the same "golden" NER dataset was split into five subsets, and for each subset an annotator was simulated with a different level of reliability σ^2 (namely, the values $\sigma^2 = \{0.005, 0.05, 0.05, 0.1, 0.1\}$ were used) according to the "model noise" method described in Section 3.4. This process was repeated 30 times and the average results for the provided testset can be found in Table 3.5. Since there was no repeated labeling, the majority voting baselines, as well as the multi-label models (MultiCRF and MultiMA-CRFX), did not apply. The obtained results indicate that, in a scenario without any repeated labeling, the proposed approach (MA-CRF) still outperforms the CRF-CONC baseline. The statistical significance of the difference between the F1-scores in the testset of these methods was evaluated through a paired t-test using Python's Scipy package[8], indicating that the difference of the means is highly significant ($p - value = 1.47 \times 10^{-11}$).

The comparison of both experiments (with and without repeated labeling) indicates that, in this setting, having less repeated labeling hurts the performance of MA-CRF. Since this model differentiates between annotators with different levels of expertise, its performance is best when the more reliable ones have annotated more sequences, which is more likely to happen with more repeated labeling. Naturally, the opposite occurs with CRF-CONC. Since in this setting the less reliable annotators dominate, more repeated labeling translates in even more predominance of

[8] http://www.scipy.org

	Trainset			Testset		
Method	Prec.	Recall	F1	Prec.	Recall	F1
MVseq	50.6%	55.6%	53.0 ± 0.4%	66.1%	63.1%	64.6 ± 2.4%
MVtoken	83.6%	76.1%	79.7 ± 0.2%	83.3%	86.9%	85.0 ± 0.7%
CRF-CONC	84.3%	84.7%	84.5 ± 1.8%	83.8%	82.9%	83.3 ± 1.9%
MultiCRF	76.6%	65.6%	70.7 ± 0.4%	75.6%	64.9%	69.8 ± 0.4%
MultiMA-CRFX	83.6%	81.3%	82.5 ± 1.0%	81.2%	79.0%	80.1 ± 1.0%
MA-CRF	**92.0%**	**91.8%**	**91.9 ± 1.9%**	**89.7%**	**89.7%**	**89.7 ± 0.8%**
CRF-GOLD	99.9%	99.9%	99.9%	95.9%	91.1%	91.0%

Table 3.6: Results for the NP chunking task with 5 simulated annotators (with $p(\text{flip}) = [0.01, 0.1, 0.3, 0.5, 0.7]$) with repeated labeling.

lower quality annotations, which affects the performance of CRF-CONC.

For the NP chunking task, the 2003 CONLL English NER dataset was also used. Besides named entities, this dataset also provides part-of-speech tags and syntactic tags (i.e. noun phrases, verbal phrases, prepositional phrases, etc.). The latter were used to generate a train and a testset for NP chunking with the same sizes of the corresponding NER datasets.

In order to simulate multiple annotators in the NP chunking data, the alternative method of randomly flipping the label of each token with uniform probability $p(\text{flip})$ was used. Since for this task there are only two possible labels for each token (part of a noun phrase or not part of a noun phrase)[9], it is trivial to simulate multiple annotators by randomly flipping labels. Using this method we simulated 5 annotators with label flip probabilities $p(\text{flip}) = \{0.01, 0.1, 0.3, 0.5, 0.7\}$. This process was repeated 30 times and the average results are presented in Table 3.6. Differently to NER, NP chunking is only a segmentation task, therefore the results for the MVseg baseline would be equal to the results for MVtoken. The experimental evidence shows that the proposed approach (MA-CRF) achieves a higher F1-score than the MVseq, MVtoken and CRF-CONC baselines. The statistical significance of the difference between the testset F1-scores of MA-CRF and all these three baselines (MVseq, MVtoken and CRF-CONC) was evaluated using a paired t-test, yielding p-values of 2×10^{-30}, 7×10^{-22} and 2×10^{-16} respectively. As with the NER task, the MA-CRF model also outperforms the MultiCRF and MultiMA-CRFX approaches (testset p-values are 6×10^{-32} and 2×10^{-21} respectively).

Amazon mechanical turk

The use of crowdsourcing platforms to annotate sequences is currently a very active research topic (Laws et al., 2011), with many different solutions being proposed to improve both the annotation and the learning processes at various levels like, for example, by evaluating annotators through the use of an expert (Voyer et al., 2010), by using a better annotation interface (Lawson et al., 2010), or by learning from partially annotated sequences thus reducing annotation costs (Fernandes and Brefeld, 2011).

[9] In fact, since a BIO decomposition is being used, there are three possible labels: B-NP, I-NP and O, and these labels are the ones that were used in the random flipping process.

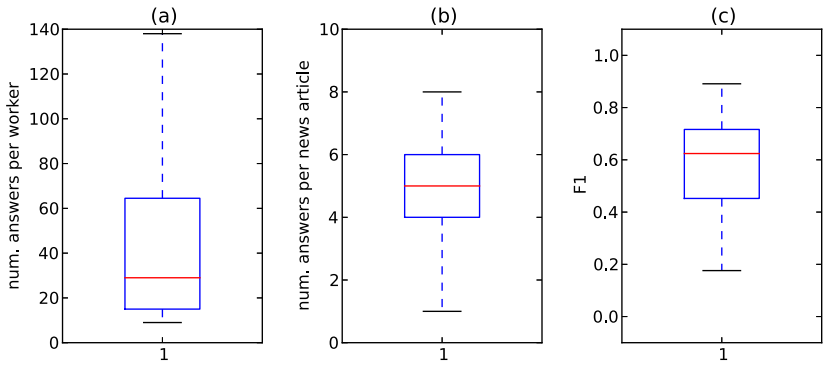


Figure 3.9: Boxplots of (a) the number of answers per AMT worker, (b) the number of answers per news article, and (c) the F1-scores of the answers provided by the different annotators.

With the purpose of obtaining real data from multiple annotators, we uploaded 400 news articles from the 2003 CONLL shared NER task (for which we have ground truth) on AMT for workers to label[10]. In this experiment, the workers were required to identify the named entities in the sentence and classify them as persons, locations, organizations or miscellaneous. Together with the named entity definition and the categories description, the workers were also provided with two exemplifying sentences. Workers with just a couple of answers were considered uninterested in the task and their answers were discarded, giving a total of 47 valid annotators. The average number of annotators per news article was 4.93, and each annotator labelled an average of 42 news articles (see Figures 3.9a and 3.9b). In order to assess the quality of the annotators, we measured the F1-scores of their answers against the ground truth. Figure 3.9c shows a boxplot of the F1-scores obtained. It is interesting to notice that the quality of the AMT workers varies enormously, with the lowest F1-score being 17.60% (a very unreliable annotator), while the highest F1-score is 89.11% (arguably almost an expert).

As with the experiments with simulated annotators, the different approaches are evaluated in the provided testset, as well as in the ground truth labels for those 400 news articles. The obtained results are presented in Table 3.7. These results indicate that the proposed approach is better at uncovering the ground truth than all the other approaches tested. This, in turn, results in a better performance on the trainset. Furthermore, we also evaluated the RMSE between the true F1-scores of the annotators (measured against the actual ground truth) and their estimated F1-scores according to the MA-CRF approach (measured against the estimated ground truth). A value of 8.61% was obtained, thus meaning that the reliability of the annotators is being approximated quite well. These results also indicate that crowdsourcing presents an interesting alternative solution for obtaining labeled data that could be used for training a NER system.

In order to evaluate the impact of repeated labeling, a random subsampling of the AMT data was performed. This experiment allows us to reproduce a situation where each article is only labeled by one annotator, thus representing the minimum cost attainable with AMT (with the same price per task). For each of the 400 news

[10]Datasets are available at: http://amilab.dei.uc.pt/fmpr/crf-ma/

	Trainset			Testset		
Method	Precision	Recall	F1	Precision	Recall	F1
MVseq	79.0%	55.2%	65.0%	44.3%	81.0%	57.3%
MVtoken	79.0%	54.2%	64.3%	45.5%	80.9%	58.2%
MVseg	83.7%	51.9%	64.1%	46.3%	82.9%	59.4%
CRF-CONC	**86.8%**	58.4%	69.8%	40.2%	**86.0%**	54.8%
MultiCRF	67.8%	15.4%	25.1%	74.8%	3.7%	7.0%
MultiMA-CRFX	79.5%	51.9%	62.8%	**84.1%**	37.1%	51.5%
MA-CRF	86.0%	**65.6%**	**74.4%**	49.4%	85.6%	**62.6%**
CRF-GOLD	99.2%	99.4%	99.3%	79.1%	80.4%	74.8%

Table 3.7: Results for the NER task using real data obtained from Amazon mechanical turk.

	Trainset			Testset		
Method	Precision	Recall	F1	Precision	Recall	F1
CRF-CONC	71.1%	42.8%	53.1 ± 10.5%	35.9%	70.1%	47.2 ± 8.7%
MA-CRF	**76.2%**	**54.2%**	**63.3 ± 1.6%**	**46.0%**	**78.2%**	**57.9 ± 1.8%**
CRF-GOLD	99.2%	99.4%	99.3%	79.1%	80.4%	74.8%

Table 3.8: Results for the NER task using data from Amazon mechanical turk without repeated labelling (subsampled data from the original dataset).

articles, a single annotator was selected at random from the set of workers who labeled that article. This process was repeated 30 times to produce 30 subsampled datasets. The average precision, recall and F1-scores of the different methods are shown in Table 3.8. Notice that, since there is no repeated labeling, both the majority voting baselines and the multi-label models (MultiCRF and MultiMA-CRFX) do not apply. The obtained results show that MA-CRF also outperforms CRF-CONC in this setting (p-value $= 3.56 \times 10^{-7}$). Interestingly, when compared to the results in Table 3.7, this experiment also shows how much could be gained by repeated labeling, thus providing a perspective on the trade-off between repeated labeling and cost.

3.5 Conclusion

In this chapter, we presented *latent expertise models*: a novel class of probabilistic models for supervised learning from multiple-annotator data. Unlike previous approaches, these models treat the reliabilities of the annotators as latent variables. This design choice results in models with various attractive characteristics, such as: its easy implementation and extension to other classifiers, the natural extension to structured prediction problems using CRFs, and the ability to overcome the overfitting to which more complex models of the annotators expertise can be susceptible as the number of instances labeled by each annotator decreases.

We empirically showed, using both simulated annotators and human-labeled data from Amazon mechanical turk, that under most conditions, the proposed MA-LR model can achieve comparable or even better results when compared to a state-of-

the-art model (Raykar et al., 2010), despite its much smaller set of parameters to model the annotators expertise. When extended to CRFs, the proposed model was shown to significantly outperform traditional approaches, such as majority voting and using the labeled data from all the annotators concatenated for training, even in situations with high levels of noise in the labels of the annotators and when the less reliable annotators dominate.

Chapter 4

Gaussian process classification with multiple annotators

4.1 Introduction

The models discussed in Chapter 3 are based on linear classifiers. This is the case with most of the state of the art in learning from multiple annotators and crowds. However, in a wide majority of classification problems, the classes are not linearly separable. For such problems, one can consider the use of basis functions as a way of achieving non-linear classification boundaries, but since the functions are fixed a-priori, the number of basis functions needed would grow exponentially with the dimensionality of the input space (Bishop, 2006). Alternatively, one can achieve non-linear classifiers by considering non-parametric models such as Gaussian processes.

In this chapter, we generalize standard Gaussian process classifiers to explicitly handle multiple annotators with different levels of expertise. Gaussian processes (GPs) are flexible non-parametric Bayesian models that fit well within the probabilistic modeling framework (Barber, 2012). By explicitly handling uncertainty, GPs provide a natural framework for properly dealing with multiple annotators with different levels of expertise. This way, we are bringing a powerful non-linear Bayesian classifier to multiple-annotator settings. Interestingly, it turns out that the computational cost of approximate Bayesian inference with expectation propagation (EP) involved in this new model is only greater up to a small factor (usually between 3 and 5) when compared with standard GP classifiers, as we shall see in Section 4.6.

Another great property of GPs is that they provide a natural framework for developing active learning strategies. Active learning is particularly important in multiple-annotator settings. Since different annotators have different levels of expertise, we wish to find both the instance whose label will be most informative for the classification model and the annotator who is more likely to provide a correct label. Aiming at reducing this cost, Chen et al. (2013) consider the problem of budget allocation in crowdsourcing environments, which they formulate as a Bayesian Markov decision process (MDP). In order to cope with computational tractability issues, they propose a new approximate policy to allocate a pre-fixed amount of budget among instance-worker pairs so that the overall accuracy can be maximized.

In the context of Gaussian processes, active learning was studied by Lawrence et al. (2003), who proposed a differential entropy score, which favours points whose inclusion leads to a large reduction in predictive (posterior) variance. This approach

was then extended by Kapoor et al. (2007), by introducing a heuristic which balances posterior mean and posterior variance. In this chapter, we propose an active learning methodology that further extends this work to multiple-annotator settings and introduces a new heuristic for selecting the best annotator to label an instance.

On a different line of work, Bachrach et al. (2012) propose a probabilistic graphical model that jointly models the difficulties of questions, the abilities of participants and the correct answers to questions in aptitude testing and crowdsourcing settings. By running approximate Bayesian inference with EP, the authors are able to query the model for the different variables of interest. Then, by exploiting the principle of entropy, the authors devise an active learning scheme, which queries the answers which are more likely to reduce the uncertainty in the estimates of the model parameters. However, this work does not address the problem of explicitly learning a classifier from multiple-annotator data as we do in this chapter. Contrarily, Yan et al. (2011) suggest an active learning methodology based on a logistic regression classifier. The authors are able to formulate the active learning problem as a bi-convex optimization problem, which they solve using a quasi-Newton numerical optimization procedure. Although this strategy is capable of jointly selecting the new training points and annotators, the fact that it requires a numerical optimization routine can make it computationally expensive. Furthermore, it is specific to the logistic-regression-based probabilistic model proposed by the authors. In this chapter, we propose a simple and yet effective active learning methodology that is well suited for the Gaussian processes framework.

The remainder of this chapter is organized as follows: Section 4.2 introduces Gaussian processes and how they can be used for regression and classification; Sections 4.3 and 4.4 describe, respectively, the proposed GP-based multiple-annotator model and how to perform inference on it; Section 4.5 describes the proposed active learning methodology; in Section 4.6 the proposed approaches are experimentally evaluated and, finally, Section 4.7 provides some conclusions.

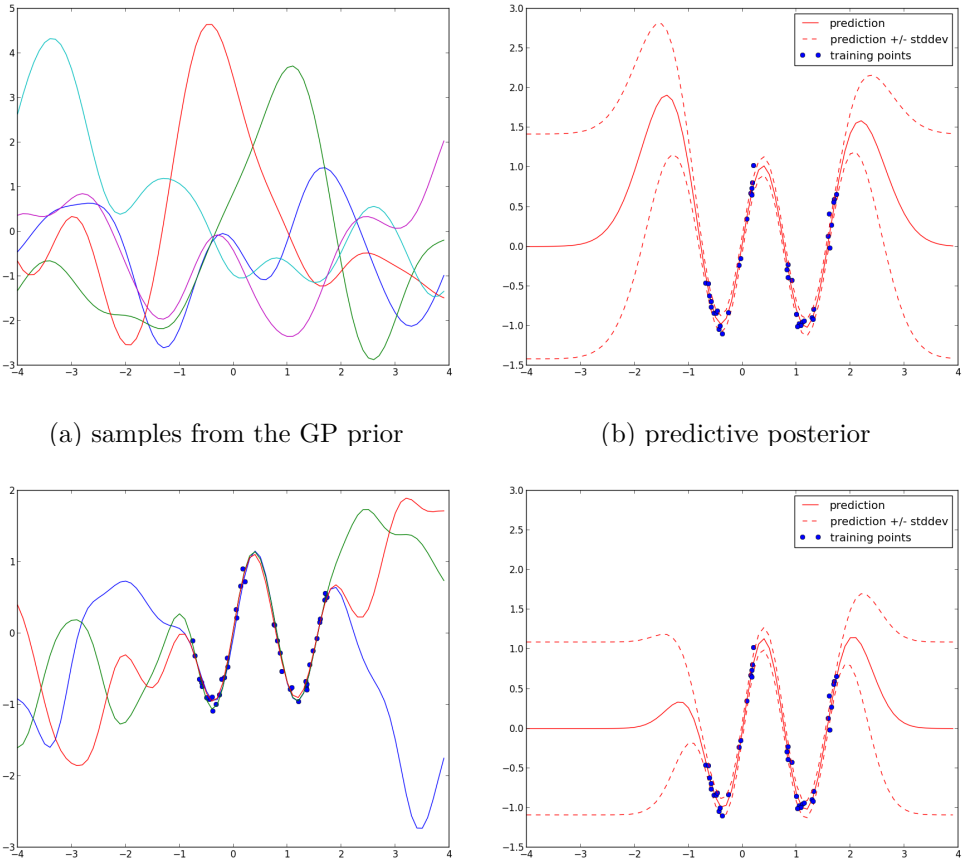
4.2 Gaussian processes

In the previous chapter, we considered linear parametric models of the inputs $\mathbf{x}$. For continuous response variables $y \in \mathbb{R}$, these models take the form

$$y = f(\mathbf{x}) + \epsilon, \tag{4.1}$$

where f is a linear parametric function of the inputs $\mathbf{x}$, such that, for example, $f(\mathbf{x}) = \boldsymbol{\eta}^{\mathrm{T}}\mathbf{x}$, and $\epsilon \sim \mathcal{N}(\epsilon|0, \sigma^2)$. This particular formulation corresponds to a linear regression model. Gaussian processes contrast with this kind of models in the sense that they are non-parametric models.

A Gaussian process (GP) is defined as a collection of random variables, any finite number of which have (consistent) joint Gaussian distributions (Rasmussen and Williams, 2005). Let us consider a multivariate (joint) Gaussian distribution, $\mathcal{N}(\mathbf{f}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$, over the N-dimensional vector $\mathbf{f} = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N))^{\mathrm{T}}$. While a multivariate Gaussian distribution is fully specified by a mean vector $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$, a GP is a stochastic process fully specified by a mean function $m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$ and a positive definite covariance function $k(\mathbf{x}, \mathbf{x}') = \mathrm{cov}[f(\mathbf{x}), f(\mathbf{x}')]$. By making use of the mean and covariance functions, GPs specify a way to determine



(a) samples from the GP prior

(b) predictive posterior

(c) samples from the GP posterior

(d) pred. post. after hyper-param. optimization

Figure 4.1: Example Gaussian process.

the mean of any arbitrary point $\mathbf{x}$ in the input space and how that point covaries with the nearby points. We can then think of GPs as a generalization of a multivariate Gaussian distribution to infinitely many variables. If we loosely see a function as a infinitely long vector $\mathbf{f}$, where each entry specifies the function value $f(\mathbf{x})$ for a particular input $\mathbf{x}$, then we can see a GP as a probability distribution over functions.

A key step in modeling data with GPs, is then to define the mean and covariance functions. The mean function defines the mean of the process and it is commonly taken to be a zero-value vector, i.e. $m(\mathbf{x}) = 0$. As for the covariance function, it specifies basics aspect of the process, such as stationarity, isotropy, smoothness and periodicity. Perhaps the most common choice of covariance function is the squared exponential, which is defined as

$$k_{\text{SE}}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{|\mathbf{x} - \mathbf{x}'|^2}{2l^2}\right), \tag{4.2}$$

with the parameter l defining the characteristic length-scale. Notice how this function goes to unity as $\mathbf{x}$ becomes closer to $\mathbf{x}'$. Hence, nearby points are more likely to covary. As a result, a GP prior with a squared exponential covariance function prefers smooth functions. Figure 4.1a shows five sample functions from a GP with zero-mean and a squared exponential covariance function with $l = 1$.

Regression

Having specified a GP prior, $p(\mathbf{f}|\mathbf{X}) = \mathcal{GP}(m(\mathbf{x}) = 0, k(\mathbf{x}, \mathbf{x}'))$, for the function values $\mathbf{f}$, where $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$ denotes the input data, the next step is to specify an appropriate likelihood function. If we are considering a regression problem, then perhaps the simplest likelihood function to use is a Gaussian distribution with mean $f(\mathbf{x})$ and σ^2 variance. Letting $\mathbf{y} = \{y_n\}_{n=1}^N$ denote the target values corresponding to the inputs $\mathbf{X}$, such that $y_n \in \mathbb{R}$, we have that $\mathbf{y} \sim \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma^2\mathbf{I}_N)$, where $\mathbf{I}_N$ refers to the $N \times N$ identity matrix. Making use of marginalization property for Gaussian distributions in (B.6), the marginal distribution of $\mathbf{y}$ is given by

$$\begin{aligned} p(\mathbf{y}|\mathbf{X}) &= \int p(\mathbf{y}|\mathbf{f})\, p(\mathbf{f}|\mathbf{X})\, d\mathbf{f} \\ &= \int \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma^2\mathbf{I}_N)\, \mathcal{N}(\mathbf{f}|\mathbf{0}_N, \mathbf{K}_N)\, d\mathbf{f} \\ &= \mathcal{N}(\mathbf{y}|\mathbf{0}_N, \underbrace{\sigma^2\mathbf{I}_N + \mathbf{K}_N}_{\mathbf{V}_N}) \\ &= \mathcal{N}(\mathbf{y}|\mathbf{0}_N, \mathbf{V}_N), \end{aligned} \tag{4.3}$$

where $\mathbf{0}_N$ is used to denote a N-dimensional vector of zeros, $\mathbf{K}_N$ denotes the covariance function $k(\mathbf{x}, \mathbf{x}')$ evaluated between every pair of training inputs, and we defined $\mathbf{V}_N \triangleq \sigma^2\mathbf{I}_N + \mathbf{K}_N$, so that $\mathbf{V}_N$ is a covariance matrix with elements

$$v(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') + \sigma^2\delta_{\mathbf{x},\mathbf{x}'}, \tag{4.4}$$

where $\delta_{\mathbf{x},\mathbf{x}'}$ denotes the Kronecker delta function, which in this case takes the value 1 if $\mathbf{x} = \mathbf{x}'$ and 0 otherwise.

In a regression problem, our aim is to make a prediction y_* for a new input $\mathbf{x}_*$. The joint distribution over $y_*, y_1, ..., y_N$ is simply given by

$$p(y_*, \mathbf{y}|\mathbf{x}_*, \mathbf{X}) = \mathcal{N}(y_*, \mathbf{y}|\mathbf{0}_{N+1}, \mathbf{V}_{N+1}), \tag{4.5}$$

where

$$\mathbf{V}_{N+1} = \begin{pmatrix} \mathbf{V}_N & \mathbf{k}_* \\ \mathbf{k}_*^{\mathrm{T}} & k_{**} + \sigma^2 \end{pmatrix}. \tag{4.6}$$

In the matrix above, we use $\mathbf{k}_*$ to denote the covariance function evaluated between the test point $\mathbf{x}_*$ and all the other training points in $\mathbf{X}$, and k_{**} to denote the covariance function evaluated between the test point $\mathbf{x}_*$ against itself, i.e. $k_{**} = k(x_*, x_*)$.

Using this joint distribution, we can now determine the distribution of y_* conditioned on $\mathbf{y}$, $\mathbf{X}$ and $\mathbf{x}_*$, i.e. the predictive distribution, by making use of the conditional probability for Gaussians from (B.5), giving:

$$p(y_*|\mathbf{y}, \mathbf{x}_*, \mathbf{X}) = \mathcal{N}(y_*|\mathbf{k}_*^{\mathrm{T}}\mathbf{V}_N\mathbf{y},\ k_{**} + \sigma^2 - \mathbf{k}_*^{\mathrm{T}}(\mathbf{V}_N)^{-1}\mathbf{k}_*). \tag{4.7}$$

Figure 4.1b shows the predictive posterior distribution given an artificial dataset consisting of 20 samples from the function $f(x) = \sin(4x)$ plus small random Gaussian white noise. Notice how the model is most confident in regions around the observed data points and becomes more and more uncertain as we get away from

those regions. Figure 4.1c shows three samples from the posterior GP. As we can see, the sampled functions are now constrained by the observations to go nearby them.

So far we have been assuming the hyper-parameter l, the length-scale of the covariance function, to be fixed. However, it can be optimized by maximizing the marginal log likelihood of the observations given by

$$\begin{aligned}\log p(\mathbf{y}|\mathbf{X}) &= \log \mathcal{N}(\mathbf{y}|\mathbf{0}_N, \mathbf{V}_N) \\ &= -\frac{D}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{V}_N| - \frac{1}{2}\mathbf{y}^{\mathrm{T}}(\mathbf{V}_N)^{-1}\mathbf{y}. \end{aligned} \tag{4.8}$$

Figure 4.1d shows the new predictive posterior distribution for the artificial dataset after optimizing the length-scale hyper-parameter l using a numeric optimizer, namely L-BFGS (Nocedal and Wright, 2006).

Classification

As we did for regression, we can use Gaussian processes for classification by choosing an appropriate likelihood function. For the sake of simplicity, we shall focus on binary classification problems, although our discussion can be generalized to multi-class problems. For binary classification problems, a common choice of likelihood functions are sigmoid-like functions such as the logistic sigmoid, which we use in Section 3.2 for logistic regression, or the probit function (Rasmussen and Williams, 2005). Here, we shall consider the probit function, which corresponds to the cumulative density function of a standard Gaussian distribution and is given by

$$\Phi(f_n) = \int_{-\infty}^{f_n} \mathcal{N}(u|0,1)\, du, \tag{4.9}$$

where, following the literature, we abbreviated $f(\mathbf{x}_n)$ as f_n, in order to simplify notation. Figure 4.2 shows the probit function for different values of f_n. This function then allows us to map the latent function values f_n into the $[0, 1]$ interval, making the values of $\Phi(f_n)$ valid probabilities which are then suitable for binary classification. The probability of an instance belonging to the positive class, $p(c_n = 1|f_n)$, then becomes $\Phi(f_n)$. Since the values of $p(c_n|f_n)$ are required to sum to 1, in order for it to be a valid probability distribution, we have that $p(c_n = 0|f_n) = 1 - p(c_n = 1|f_n) = \Phi(-f_n)$, where we made use of the fact that $1 - \Phi(f_n) = \Phi(-f_n)$. The likelihood can then be written as $p(c_n|f_n) = \Phi((-1)^{(1-c_n)} f_n)$, which can easily be verified by assigning values to $c_n \in \{0, 1\}$. Gaussian process classification then proceeds by placing a GP prior over the latent function f. Figure 4.3 shows a factor graph representation of the GP model for classification.

In order to predict the class c_* of a new test point $\mathbf{x}_*$ given an observed dataset $\mathcal{D} = \{\mathbf{X}, \mathbf{c}\}$, we first compute the distribution of the latent variable f_* corresponding to the test point $\mathbf{x}_*$

$$p(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{c}) = \int p(f_*|\mathbf{f}, \mathbf{x}_*, \mathbf{X})\, p(\mathbf{f}|\mathbf{X}, \mathbf{c})\, d\mathbf{f}, \tag{4.10}$$

and then use this distribution to compute the predictive distribution

$$p(c_* = 1|\mathbf{x}_*, \mathbf{X}, \mathbf{c}) = \int \Phi(f_*)\, p(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{c})\, df_*. \tag{4.11}$$

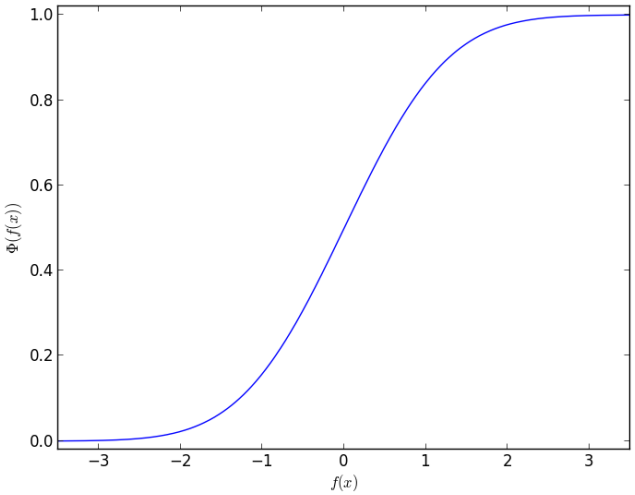


Figure 4.2: Probit function.

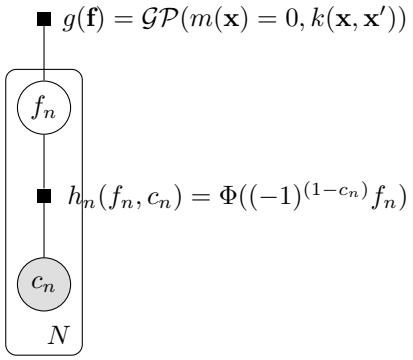


Figure 4.3: Factor graph for GP classification.

For a probit likelihood, the predictive distribution in (4.11) can be easily evaluated analytically. However, the distribution over f_* given by (4.10) is now intractable to compute, due to the fact that the posterior over the latent functions values $p(\mathbf{f}|\mathbf{X}, \mathbf{c})$ is non-Gaussian. This a consequence of the fact that the probit likelihood is not conjugate to the Gaussian process prior. Hence, we cannot simply apply Bayes theorem in order to obtain an exact answer as follows

$$\overbrace{p(\mathbf{f}|\mathbf{X}, \mathbf{c})}^{\text{posterior}} = \frac{\overbrace{p(\mathbf{c}|\mathbf{f})}^{\text{probit likelihood}} \overbrace{p(\mathbf{f}|\mathbf{X})}^{\text{GP prior}}}{\underbrace{p(\mathbf{c}|\mathbf{X})}_{\text{model evidence}}}. \tag{4.12}$$

A standard procedure is to use EP to approximate this posterior distribution with a Gaussian. The interested reader is redirected to (Rasmussen and Williams, 2005) for the details on the EP algorithm for GP classification with a probit likelihood. In the following section, we extend this model to multiple-annotator settings.

4.3 Proposed model

As previously discussed, when learning how to classify from multiple annotators, instead of a single true class label c_n for the n^{th} instance, we are given a vector of class labels $\mathbf{y}_n = (y_n^1, ..., y_n^R)^{\mathrm{T}}$, corresponding to the noisy labels provided by the R annotators that labeled that instance. Hence, a dataset $\mathcal{D}$ of size N is defined as $\mathcal{D} = \{\mathbf{X}, \mathbf{Y}\}$, where $\mathbf{Y} = (\mathbf{y}_1, ..., \mathbf{y}_N)^{\mathrm{T}}$.

Letting c_n be the unobserved true class label for a given input point $\mathbf{x}_n$, our goal is to estimate the posterior distribution of c_* for a new test point $\mathbf{x}_*$. Mathematically, we want to compute

$$p(c_* = 1|\mathbf{x}_*, \mathbf{X}, \mathbf{Y}) = \int \Phi(f_*)\, p(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{Y})\, df_*, \tag{4.13}$$

where the posterior distribution of the latent variable f_* is given by the following integral

$$p(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{Y}) = \int p(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{f})\, p(\mathbf{f}|\mathbf{X}, \mathbf{Y})\, d\mathbf{f}. \tag{4.14}$$

So far, these two equations are very similar to the ones for standard GP classification, i.e. (4.11) and (4.10). As we did there, we shall place a GP prior on $\mathbf{f}$, such that $\mathbf{f}|\mathbf{X} \sim \mathcal{GP}(m(\mathbf{x}) = 0, k(\mathbf{x}, \mathbf{x}'))$. By making use of Bayes rule, the posterior distribution of the latent variables $p(\mathbf{f}|\mathbf{X}, \mathbf{Y})$ that appears on the right-hand side of (4.14) becomes

$$p(\mathbf{f}|\mathbf{X}, \mathbf{Y}) = \frac{p(\mathbf{f}|\mathbf{X})\, p(\mathbf{Y}|\mathbf{f})}{p(\mathbf{Y}|\mathbf{X})}, \tag{4.15}$$

where the prior $p(\mathbf{f}|\mathbf{X})$ is a zero-mean Gaussian distribution $\mathcal{N}(\mathbf{f}|\mathbf{0}_N, \mathbf{K}_N)$ with a $N \times N$ covariance matrix $\mathbf{K}_N$ obtained by evaluating the covariance function $k(\mathbf{x}, \mathbf{x}')$ between all input points, $p(\mathbf{Y}|\mathbf{f})$ is the likelihood term, and the denominator $p(\mathbf{Y}|\mathbf{X})$ corresponds to the marginal likelihood of the data.

So far, we have not established how to model $p(\mathbf{Y}|\mathbf{f})$. In order to do that, we make use of the latent variable c introduced earlier, which corresponds to the (latent) true class labels. Using this latent variable, we can define the data-generating process to be the following: for each input point $\mathbf{x}_n$ there is a (latent) true class label c_n, and the different R annotators then provide noisy versions y_n^r of c_n. This amounts to saying that $p(\mathbf{y}_n|f_n) = \sum_{c_n} p(c_n|f_n)\, p(\mathbf{y}_n|c_n)$. Assuming that the annotators make their decisions independently of each other allows $p(\mathbf{y}_n|c_n)$ to factorize, yielding

$$p(\mathbf{y}_n|f_n) = \sum_{c_n} p(c_n|f_n) \prod_{r=1}^{R} p(y_n^r|c_n), \tag{4.16}$$

where $p(c_n|f_n) = \Phi((-1)^{(1-c_n)} f_n)$ is the probit likelihood for values of $c_n \in \{0, 1\}$, and

$$y_n^r|c_n = 1 \sim Bernoulli(\alpha^r)$$
$$y_n^r|c_n = 0 \sim Bernoulli(1 - \beta^r).$$

The parameters of these Bernoullis, α^r and β^r, can therefore be interpreted as the sensitivity and specificity, respectively, of the r^{th} annotator. This formulation is similar to one proposed by Raykar et al. (2010). As we discussed in the previous

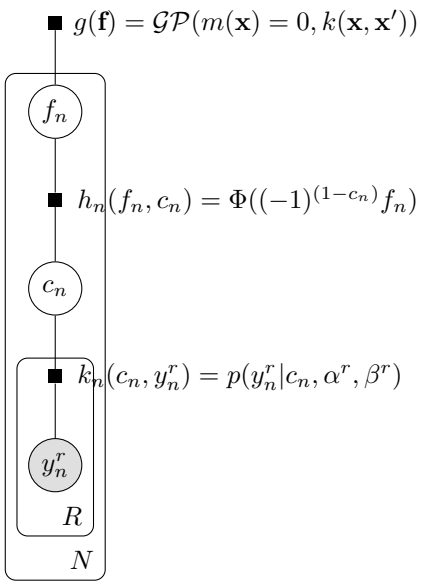


Figure 4.4: Factor graph for GP classification with multiple annotators.

chapter, this approach has the advantage of being able to capture biases in the annotators' labeling style. Figure 4.4 shows a factor graph representation of the proposed multiple-annotator GP classification model, where the differences to the factor graph for standard GP classification in Figure 4.3 become clear. Also, please notice that in situations where each annotator does not label all the instances, R can be simply replaced by R_n, which denotes the annotators that labeled the n^{th} instance.

Since the values of c are not observed, we have to marginalize over them by summing over all its possible values. Hence,

$$p(\mathbf{f}|\mathbf{X}, \mathbf{Y}) = \frac{p(\mathbf{f}|\mathbf{X}) \sum_{\mathbf{c}} p(\mathbf{Y}|\mathbf{c})\, p(\mathbf{c}|\mathbf{f})}{p(\mathbf{Y}|\mathbf{X})}, \tag{4.17}$$

where we introduced the vector $\mathbf{c} = (c_1, ..., c_N)^{\mathrm{T}}$.

By making use of the i.i.d. assumption of the data, we can re-write the posterior of the latent variables $\mathbf{f}$ in (4.15) as

$$p(\mathbf{f}|\mathbf{X}, \mathbf{Y}) = \frac{1}{Z}\, p(\mathbf{f}|\mathbf{X}) \prod_{n=1}^{N} \sum_{c_n \in \{0,1\}} p(\mathbf{y}_n|c_n)\, p(c_n|f_n), \tag{4.18}$$

where Z is a normalization constant corresponding to the marginal likelihood of the data $p(\mathbf{Y}|\mathbf{X})$. As with standard GP classification, the non-Gaussian likelihood term deems the posterior distribution of the latent variables $p(\mathbf{f}|\mathbf{X}, \mathbf{Y})$ also non-Gaussian, thus making the integral in (4.14) intractable. In the following section, we shall develop an EP algorithm (Minka, 2001) for performing approximate Bayesian inference in this model.

4.4 Approximate inference

Our goal with EP is to approximate the posterior distribution of the latent variables $p(\mathbf{f}|\mathbf{X}, \mathbf{Y})$ with a Gaussian distribution $q(\mathbf{f}|\mathbf{X}, \mathbf{Y}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. In EP, we

approximate the likelihood terms by local likelihood approximations in the form of unnormalized Gaussian functions in the latent variables f_n

$$\sum_{c_n \in \{0,1\}} p(\mathbf{y}_n|c_n)\, p(c_n|f_n) \simeq t_n(f_n|\tilde{Z}_n, \tilde{\mu}_n, \tilde{\sigma}_n^2) \triangleq \tilde{Z}_n \mathcal{N}(f_n|\tilde{\mu}_n, \tilde{\sigma}_n^2), \tag{4.19}$$

which defines the site parameters $\tilde{Z}_n$, $\tilde{\mu}_n$ and $\tilde{\sigma}_n^2$ of EP.

Also, in EP we abandon exact normalization for tractability. The product of the (independent) likelihoods t_n is then given by (Rasmussen and Williams, 2005)

$$\prod_{n=1}^{N} t_n(f_n|\tilde{Z}, \tilde{\mu}_n, \tilde{\sigma}_n^2) = \mathcal{N}(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Sigma}}) \prod_{n=1}^{N} \tilde{Z}_n, \tag{4.20}$$

where $\tilde{\boldsymbol{\mu}}$ is a vector of $\tilde{\mu}_n$ and $\tilde{\boldsymbol{\Sigma}}$ is a diagonal matrix with $\tilde{\boldsymbol{\Sigma}}_{nn} = \tilde{\sigma}_n^2$.

The posterior $p(\mathbf{f}|\mathbf{X}, \mathbf{Y})$ is then approximated by $q(\mathbf{f}|\mathbf{X}, \mathbf{Y})$, which is given by

$$\begin{aligned} q(\mathbf{f}|\mathbf{X}, \mathbf{Y}) &\triangleq \frac{1}{Z_{EP}} p(\mathbf{f}|\mathbf{X}) \prod_{n=1}^{N} t_n(f_n|\tilde{Z}, \tilde{\mu}_n, \tilde{\sigma}_n^2) \\ &= \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \end{aligned} \tag{4.21}$$

with $\boldsymbol{\mu} = \boldsymbol{\Sigma}\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{\mu}}$ and $\boldsymbol{\Sigma} = (\mathbf{K}_N^{-1} + \tilde{\boldsymbol{\Sigma}}^{-1})^{-1}$, where we made use of the formula for the product of two Gaussians from Eq. B.1. The normalization constant, $Z_{EP} = q(\mathbf{Y}|\mathbf{X})$, is the EP algorithm's approximation to the normalization term Z used in (4.18).

All there is to do now, is to choose the parameters of the local approximating distributions t_n. In EP, this consists of three steps. In step 1, the cavity distribution $q_{-n}(f_n)$ is computed by making use of the result in Eq. B.3 for the division of two Gaussians to divide the approximate posterior marginal $q(f_n|\mathbf{X}, \mathbf{Y}) = \mathcal{N}(f_n|\mu_n, \sigma_n)$ by the approximate likelihood term t_n that we want to refine, yielding

$$\begin{aligned} q_{-n}(f_n) &\propto \int p(\mathbf{f}|\mathbf{X}) \prod_{j \neq n}^{N} t_j(f_j, \tilde{Z}_j, \tilde{\mu}_j, \tilde{\sigma}_j^2)\, df_j \\ &\triangleq \mathcal{N}(f_n|\mu_{-n}, \sigma_{-n}^2), \end{aligned} \tag{4.22}$$

where

$$\mu_{-n} = \sigma_{-n}^2(\sigma_n^{-2}\mu_n - \tilde{\sigma}_n^{-2}\tilde{\mu}_n) \tag{4.23}$$

$$\sigma_{-n}^2 = (\sigma_n^{-2} - \tilde{\sigma}_n^{-2})^{-1}. \tag{4.24}$$

In step 2, we combine the cavity distribution with the exact likelihood term, $\sum_{c_n \in \{0,1\}} p(\mathbf{y}_n|c_n) p(c_n|f_n)$, to get the desired (non-Gaussian) marginal, given by

$$\begin{aligned} \hat{q}(f_n) &\triangleq \hat{Z}_n \mathcal{N}(\hat{\mu}_n, \hat{\sigma}_n^2) \\ &\simeq q_{-n}(f_n) \sum_{c_n \in \{0,1\}} p(\mathbf{y}_n|c_n)\, p(c_n|f_n). \end{aligned} \tag{4.25}$$

By making use of the definitions of $p(\mathbf{y}_n|c_n)$ and $p(c_n|f_n)$ introduced earlier, this expression can be further manipulated, giving

$$\begin{aligned} \hat{q}(f_n) &\simeq q_{-n}(f_n)(1 - \Phi(f_n)) \prod_{r=1}^{R} p(y_n^r|c_n = 0) + q_{-n}(f_n)\, \Phi(f_n) \prod_{r=1}^{R} p(y_n^r|c_n = 1) \\ &= b_n \mathcal{N}(f_n|\mu_{-n}, \sigma_{-n}^2) + (a_n - b_n)\, \Phi(f_n)\, \mathcal{N}(f_n|\mu_{-n}, \sigma_{-n}^2), \end{aligned} \tag{4.26}$$

where we defined

$$a_n = \prod_{r=1}^{R} p(y_n^r | c_n = 1) = \prod_{r=1}^{R} (\alpha^r)^{(y_n)} (1 - \alpha^r)^{(1-y_n)} \tag{4.27}$$

$$b_n = \prod_{r=1}^{R} p(y_n^r | c_n = 0) = \prod_{r=1}^{R} (1 - \beta^r)^{(y_n)} (\beta^r)^{(1-y_n)}. \tag{4.28}$$

We then choose a Gaussian approximation to the non-Gaussian marginal in (4.26) by moment matching, i.e. we pick the Gaussian approximation that matches the moments of (4.26). These moments are given by

$$\hat{Z}_n = b_n + (a_n - b_n)\, \Phi(\eta_n) \tag{4.29}$$

$$\hat{\mu}_n = \mu_{-n} + \frac{(a_n - b_n)\, \sigma_{-n}^2\, \mathcal{N}(\eta_n)}{\Big[b_n + (a_n - b_n)\, \Phi(\eta_n)\Big] \sqrt{1 + \sigma_{-n}^2}} \tag{4.30}$$

$$\hat{\sigma}_n^2 = \sigma_{-n}^2 - \frac{\sigma_{-n}^4}{1 + \sigma_{-n}^2} \left(\frac{\eta_n\, \mathcal{N}(\eta_n)\, (a_n - b_n)}{b_n + (a_n - b_n)\, \Phi(\eta_n)} + \frac{\mathcal{N}(\eta_n)^2\, (a_n - b_n)^2}{(b_n + (a_n - b_n)\, \Phi(\eta_n))^2} \right), \tag{4.31}$$

where

$$\eta_n \triangleq \frac{\mu_{-n}}{\sqrt{1 + \sigma_{-n}^2}}.$$

The derivation of these moments can be found in Appendix C.1. Notice how, in the particular case when $R = 1$, $\alpha^r = 1$ and $\beta^r = 1$, we get back the moments for the standard GP classification model with a probit likelihood (see Rasmussen and Williams (2005)). This shows how the proposed model is a generalization of the standard GP classification model to multiple annotators, having the standard single-annotator version as a special case.

Finally, in step 3, we compute the approximations t_n that make the posterior have the desired marginals from step 2. Particularly, we want the product of the cavity distribution and the local approximation to have the desired moments, leading to (Rasmussen and Williams, 2005)

$$t_n(f_n | \tilde{Z}_n, \tilde{\mu}_n, \tilde{\sigma}_n^2) = \frac{\hat{q}(f_n)}{q_{-n}(f_n)} \tag{4.32}$$

$$\tilde{\mu}_n = \tilde{\sigma}_n^2 (\hat{\sigma}_n^{-2} \hat{\mu}_n - \sigma_{-n}^{-2} \mu_{-n}) \tag{4.33}$$

$$\tilde{\sigma}_n^2 = (\hat{\sigma}_n^{-2} - \sigma_{-n}^{-2})^{-1} \tag{4.34}$$

$$\tilde{Z}_n = \hat{Z}_n \sqrt{2\pi} \sqrt{\sigma_{-n}^2 + \tilde{\sigma}_n^2}\, \exp\left(\frac{1}{2} \frac{\mu_{-n} - \tilde{\mu}_n}{\sigma_{-n}^2 - \tilde{\sigma}_n^2} \right), \tag{4.35}$$

where we made use of the formula for the division of two Gaussians in Eq. B.3.

The different local approximating terms t_n are then updated sequentially by iterating through these three steps until convergence.

In order to make predictions, we make use of the EP approximation to the posterior distribution $q(\mathbf{f}|\mathbf{X}, \mathbf{Y})$ defined in (4.21), and plug it in (4.14) to compute the predictive mean and variance of the latent variable f_*

$$\mathbb{E}_q[f_* | \mathbf{x}_*, \mathbf{X}, \mathbf{Y}] = \mathbf{k}_*^{\mathrm{T}} (\mathbf{K}_N + \tilde{\boldsymbol{\Sigma}})^{-1} \tilde{\boldsymbol{\mu}} \tag{4.36}$$

$$\mathbb{V}_q[f_* | \mathbf{x}_*, \mathbf{X}, \mathbf{Y}] = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^{\mathrm{T}} (\mathbf{K}_N + \tilde{\boldsymbol{\Sigma}})^{-1} \mathbf{k}_*, \tag{4.37}$$

where $\mathbf{k}_*$ is a vector whose entries correspond to the covariance function $k(\mathbf{x}, \mathbf{x}')$ evaluated between the test point $\mathbf{x}_*$ and all the training input points.

Finally, the approximate predictive distribution for the true class label c_* is given by the integral in (4.13), which can be analytically approximated as (Rasmussen and Williams, 2005)

$$q(c_* = 1|\mathbf{x}_*, \mathbf{X}, \mathbf{Y}) = \Phi\left(\frac{\mathbf{k}_*^{\mathrm{T}}(\mathbf{K}_N + \tilde{\boldsymbol{\Sigma}})^{-1}\tilde{\boldsymbol{\mu}}}{\sqrt{1 + k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^{\mathrm{T}}(\mathbf{K}_N + \tilde{\boldsymbol{\Sigma}})^{-1}\mathbf{k}_*}}\right). \tag{4.38}$$

So far, we have been assuming the annotators' parameters α^r and β^r to be fixed. However, we need to estimate those as well. This is done iteratively by scheduling the updates as follows: every i EP sweeps through the data, or alternatively, when the difference in the marginal likelihood between two consecutive iterations ϵ falls below a certain threshold[1], the values of α^r and β^r are re-estimated as

$$\alpha^r = \frac{\sum_{n=1}^{N} y_n^r \, q(c_n = 1|\mathbf{X}, \mathbf{Y})}{\sum_{n=1}^{N} q(c_n = 1|\mathbf{X}, \mathbf{Y})} \tag{4.39}$$

$$\beta^r = \frac{\sum_{n=1}^{N} (1 - y_n^r)(1 - q(c_n = 1|\mathbf{X}, \mathbf{Y}))}{\sum_{n=1}^{N} 1 - q(c_n = 1|\mathbf{X}, \mathbf{Y})}. \tag{4.40}$$

Although this will raise the computational cost of EP, as we shall see in Section 4.6, this increase is only by a small factor.

4.5 Active learning

The full Bayesian treatment of the Gaussian process framework provides natural extensions to active learning settings, which can ultimately reduce the annotation cost even further.

In active learning with multiple annotators our goal is twofold: (1) pick an instance to label next and (2) pick the best annotator to label it. For simplicity, we choose to treat the two problems separately. Hence, in order to pick an instance to label, we take the posterior distribution of the latent variable $p(f_u|\mathbf{x}_u, \mathbf{X}, \mathbf{Y}) = \mathcal{N}(f_u|\mu_u, \sigma_u^2)$ for all unlabeled data points $\mathbf{x}_u \in \mathbf{X}_u$ and compute

$$\mathbf{x}_* = \arg\min_{\mathbf{x}_u \in \mathbf{X}_u} \frac{|\mu_u|}{\sqrt{1 + \sigma_u}}. \tag{4.41}$$

This approach is analogous to the one proposed in Kapoor et al. (2007) for single-annotator settings and provides a balance between the distance to the decision boundary, given by the posterior mean $|\mu_u|$, and the posterior variance σ_u (uncertainty) associated with that point.

As for the choice of the annotator to label the instance picked, we proceed by identifying the annotator who is more likely to label it correctly given our current state of knowledge, i.e. given our prior beliefs of the class which the instance belongs to and the information about the levels of expertise of the different annotators.

[1] During the experiments, these values were set to $i = 3$ and $\epsilon = 10^{-4}$.

Mathematically, we want to pick the annotator r_* that maximizes

$$r_* = \arg\max_r \Big(p(y^r = 1|c = 1)\, q(c = 1|\mathbf{x}_*, \mathbf{X}, \mathbf{Y}) + p(y^r = 0|c = 0)\, q(c = 0|\mathbf{x}_*, \mathbf{X}, \mathbf{Y}) \Big)$$
$$= \arg\max_r \Big(\alpha^r q(c = 1|\mathbf{x}_*, \mathbf{X}, \mathbf{Y}) + \beta^r \big(1 - q(c = 1|\mathbf{x}_*, \mathbf{X}, \mathbf{Y})\big) \Big). \quad (4.42)$$

However, since we are now actively picking the annotators, there is a risk of generating a model that is biased towards labels from a single annotator when using this heuristic. This happens because, if a single annotator provides the majority of the labels, the estimate of the ground truth will be biased towards her opinion. Consequently, her sensitivity and specificity parameters will also be biased, and she might end up being selected over and over. In order to address this issue, we introduce a dependency on the annotator r when estimating α^r and β^r. Namely, we replace $q(c = 1|\mathbf{X}, \mathbf{Y})$ with $q(c = 1|\mathbf{X}\backslash\mathbf{x}^r, \mathbf{Y}\backslash\mathbf{y}^r)$ in (4.39) and (4.40), where $\mathbf{Y}\backslash\mathbf{y}^r$ denotes all the labels except the ones from annotator r, thereby deeming the ground truth estimates used for computing the reliability parameters α^r and β^r of annotator r, independent of her own answers.

4.6 Experiments

The proposed approaches[2] are validated using both real and simulated annotators on real datasets from different application domains.

Simulated annotators

In order to simulate annotators with different levels of expertise, we start by assigning a sensitivity α^r and specificity β^r to each of the simulated annotators. Then for each training point, we simulate the answer of the r^{th} annotator by sampling y_i^r from a Bernoulli(α^r) if the training point belongs to the positive class, and by sampling y_i^r from Bernoulli($1 - \beta^r$) otherwise. This way, we can simulate annotators whose expected values for the sensitivity and specificity will tend to α^r and β^r respectively, as the number of training points goes to infinity.

This annotator simulation process is applied to various datasets from the UCI repository[3], and the results of the proposed approach (henceforward referred to as GPC-MA) is compared with two baselines: one consisting of using the majority vote for each instance (referred as GPC-MV), and another baseline consisting of using all data points from all annotators as training data (GPC-CONC). Note that if we simulate 7 annotators, then the dataset for the latter baseline will be 7 times larger than the former one. In order to also provide an upper bound/baseline we also show the results of a Gaussian process classifier applied to the true (golden) labels $\mathbf{c}$ (referred as GPC-GOLD).

Table 4.1 shows the results obtained in 6 UCI datasets, by simulating 7 annotators with sensitivities $\alpha = \{0.9, 0.9, 0.8, 0.4, 0.3, 0.4, 0.6, 0.5\}$ and specificities $\beta = \{0.8, 0.9, 0.9, 0.4, 0.5, 0.5, 0.5, 0.4\}$. For all experiments, a random 70/30 train/test split was performed and an isotropic squared exponential covariance function was

[2]Source code and datasets are available at: http://amilab.dei.uc.pt/fmpr/gpc-ma/
[3]http://archive.ics.uci.edu/ml/

		Trainset		Testset	
	Method	Acc.	AUC	Acc.	AUC
ionosphere	GPC-GOLD	1.000	1.000	0.900	0.999
	GPC-CONC	0.811	0.880	0.743	0.830
	GPC-MV	0.726	0.853	0.693	0.708
	GPC-MA	**0.978**	**0.998**	**0.889**	**0.987**
pima	GPC-GOLD	1.000	1.000	0.993	1.000
	GPC-CONC	0.848	0.900	0.860	0.930
	GPC-MV	0.840	0.955	0.860	0.967
	GPC-MA	**0.994**	**1.000**	**0.991**	**1.000**
parkinsons	GPC-GOLD	1.000	1.000	0.992	0.999
	GPC-CONC	0.827	0.889	0.851	0.899
	GPC-MV	0.663	0.895	0.692	0.867
	GPC-MA	**0.910**	**0.999**	**0.947**	**0.992**
bupa	GPC-GOLD	1.000	1.000	0.993	1.000
	GPC-CONC	0.862	0.926	0.854	0.932
	GPC-MV	0.793	0.961	0.816	0.953
	GPC-MA	**0.995**	**1.000**	**0.991**	**1.000**
breast	GPC-GOLD	1.000	1.000	0.997	1.000
	GPC-CONC	0.922	0.938	0.936	0.983
	GPC-MV	0.860	0.990	0.887	0.992
	GPC-MA	**0.995**	**1.000**	**0.996**	**1.000**
tic-tac-toe	GPC-GOLD	1.000	1.000	1.000	1.000
	GPC-CONC	0.828	0.887	0.884	0.952
	GPC-MV	0.717	0.932	0.806	0.958
	GPC-MA	**0.999**	**1.000**	**1.000**	**1.000**

Table 4.1: Average accuracy and AUC over 30 runs, obtained by simulating 7 artificial annotators on different UCI datasets.

used. Taking advantage of the stochastic nature of the annotators' simulation process, we repeat each experiment 30 times and always report the average results along with with respective standard deviations. Besides testset results, we also report performance metrics on the trainset because this corresponds to the important problem of uncovering the ground truth labels from the noisy answers of multiple annotators. The statistical significance of the differences between GPC-MA and the best baseline method was evaluated using a paired t-test, yielding a p-value smaller than 2.2×10^{-16} for all datasets.

In order to compare the different approaches in terms of computational demands, the execution times were also measured. Table 4.2 shows the average execution times over 30 runs on a Intel Core i7 2600 (3.4GHZ) machine with 32GB DDR3 (1600MHZ) of memory.

The results obtained show that the proposed approach (GPC-MA) consistently outperforms the two baselines in the 6 datasets used, while only raising the computational time by a small factor (between 3 and 5) when compared to the majority voting baseline. Furthermore, we can see that GPC-MA is considerably faster (up to 100x) than the GPC-CONC baseline, which is not surprising since the computational complexity of GPs is $\mathcal{O}(N^3)$ and the dataset used in GPC-CONC is R-times larger than the original dataset. However, GPC-CONC seems to perform better

Dataset	GOLD	CONC	MV	GPC-MA
ionosphere	0.495	403.618	0.476	2.470
pima	0.551	357.238	0.445	2.583
parkinsons	0.187	55.424	0.186	0.608
bupa	0.551	357.238	0.445	2.583
breast	2.176	3071.467	1.474	8.093
tic-tac-toe	3.67	5035.112	3.106	16.130

Table 4.2: Average execution times (in seconds) over 30 runs.

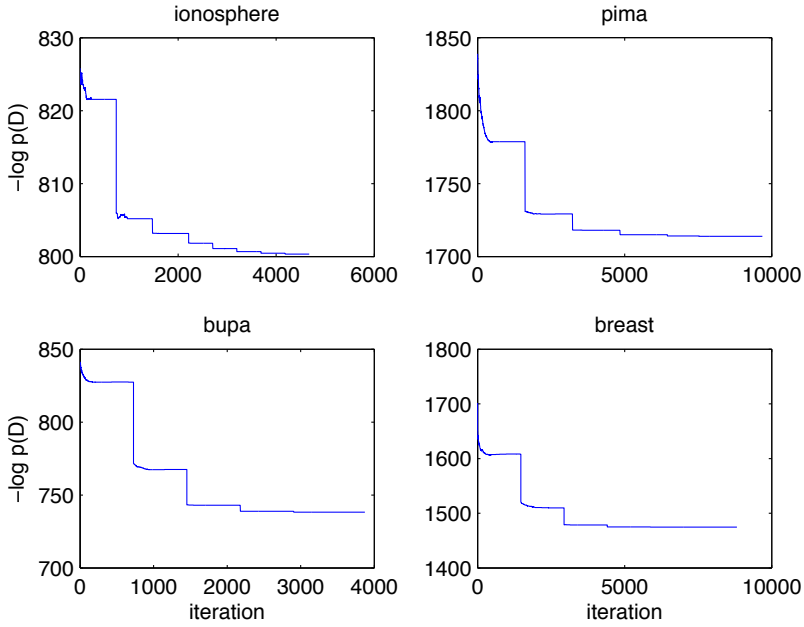


Figure 4.5: Plots of the log marginal likelihood over 4 runs of GPC-MA using 4 different datasets.

than the other baseline method: GPC-MV. We hypothesize that this is due to the fact that GPC-CONC can model the uncertainty introduced by the heterogeneity in the annotators' answers, by considering how much these vary in a certain region of the space, while the GPC-MV aggregates all the answers regardless of how consistent they are. Hence, if for example, all 7 annotators assign the same label to some data point, the variance associated with that data point will be lower than when the 7 annotators provide contradicting labels.

Figure 4.5 shows plots of the (negative) log marginal likelihood over 4 runs of GPC-MA using 4 different datasets, where it becomes clear the effect of the re-estimation of the annotator's parameters α and β, which is evidenced by the periodic "steps" in the log marginal likelihood.

Real annotators

The proposed approach was also evaluated on real multiple-annotator settings by applying it to the sentiment polarity and a music genre classification datasets used in Section 3.4.

	Trainset		Testset	
Method	Accuracy	AUC	Accuracy	AUC
GPC-GOLD	0.987	0.999	0.723	0.785
GPC-MV	0.886	0.923	0.719	0.781
GPC-MA	**0.900**	**0.944**	**0.721**	**0.783**

Table 4.3: Results for the sentiment polarity dataset.

	Trainset		Testset	
Method	AUC	F1	AUC	F1
GPC-GOLD	1.000	1.000	0.852	0.683
GPC-CONC	0.926	0.700	0.695	0.423
GPC-MV	0.812	0.653	0.661	0.411
GPC-MA	**0.943**	**0.702**	**0.882**	**0.601**

Table 4.4: Results obtained for the music genre dataset.

Tables 4.3 and 4.4 show the results obtained for the different approaches in the sentiment and music datasets respectively. Since the music dataset corresponds to a multi-class problem, we proceeded by transforming it into 10 different binary classification tasks. Hence, each task corresponds to identifying songs of each genre. Unlike the previous experiments, with the music genre dataset a squared exponential covariance function with automatic relevance determination (ARD) was used, and the hyper-parameters were optimized by maximizing the marginal likelihood.

Due to the computational cost of GPC-CONC and the size of the sentiment dataset, we were unable to test this method on this dataset. Nevertheless, the obtained results show the overall advantage of GPC-MA over the baseline methods.

Active learning

The active learning heuristics proposed were tested on the music genre dataset from Section 4.6. For each genre, we randomly initialize the algorithm with 200 instances and then perform active learning for another 300 instances. In order to make active learning more efficient, in each iteration we rank the unlabeled instances according to (4.41) and select the top 10 instances to label. For each of these instances we query the best annotator according to the heuristic we proposed for selecting annotators (4.42). Since each instance in the dataset is labeled by an average of 4.21 annotators, picking a single annotator per instance corresponds to savings in annotation cost of more than 76%. Each experiment is repeated 30 times with different random initializations. Figure 4.6 shows how the average testset AUC for the different music genres evolves as more labels are queried. We compare the proposed active learning methodology with a random baseline. In order to make clear the individual contributions of each of the heuristics proposed, we also show the results of using only the heuristic in (4.41) for selecting an instance to label and selecting the annotators at random. As the figure evidences, there is a clear advantage in using both active learning heuristics together, which can provide an improvement in AUC of more than 10% after the 300 queries.

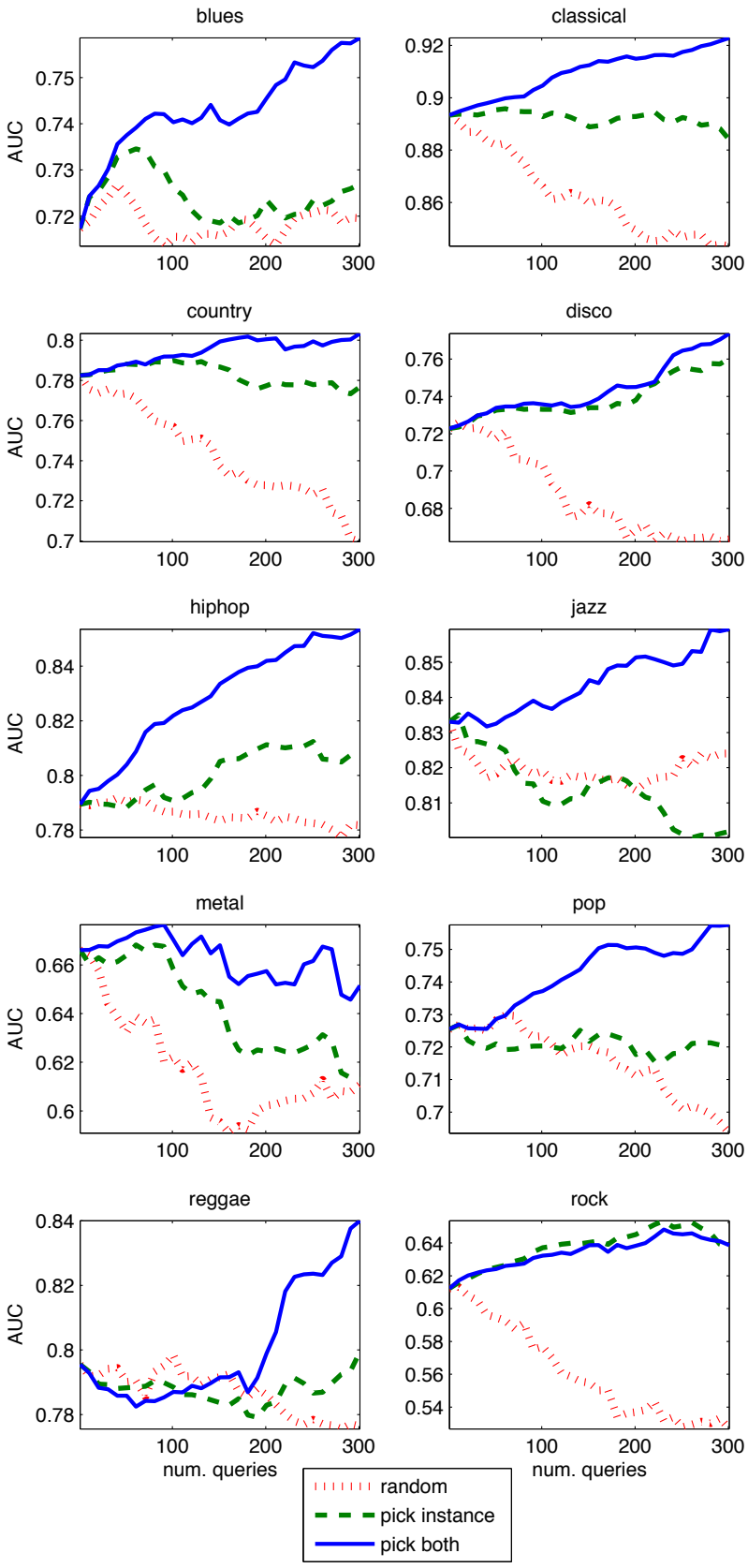


Figure 4.6: Active learning results on music genre dataset.

4.7 Conclusion

In this chapter, we presented a non-linear non-parametric Bayesian classifier for multiple-annotator settings, which corresponds to a generalization of the Gaussian process classifier (a special case when $R = 1$, $\alpha = 1$ and $\beta = 1$). By treating the unobserved true labels as latent variables, this model is able to estimate the different levels of expertise of the multiple annotators, thereby being able to compensate for their biases and thus obtaining better estimates of the ground truth labels. We empirically show, using both simulated annotators and real multiple-annotator data collected from Amazon mechanical turk, that while this model only incurs in a small increase in the computational cost of approximate Bayesian inference with EP, it is able to significantly outperform all the baseline methods. Furthermore, two simple and yet effective active learning heuristics were proposed, which can provide an even further boost in classification performance, while reducing the number of annotations required, and consequently the annotation cost.

Chapter 5

Learning supervised topic models from crowds

5.1 Introduction

So far, we have been assuming the inputs $\mathbf{x}$ of our supervised learning models to be feature vectors resultant from some feature extraction procedure or another kind of pre-processed phase. Hence, in the previous chapters, no models of $p(\mathbf{x})$ were considered. However, in many situations, we deal with complex high-dimensional data, such as images or text. In these cases, working with the raw data is often not the best approach. A common solution is to use topic models. In fact, the growing need to analyze large document corpora has led to great developments in topic modeling. Topic models, such as latent Dirichlet allocation (LDA) (Blei et al., 2003), allow us to analyze large collections of documents, by revealing their underlying themes, or topics, and how each document exhibits them. Therefore, it is not surprising that topic models have become a standard tool in data analysis and machine learning, with many applications that transcend their original purpose of modeling textual data, such as analyzing images (Fei-Fei and Perona, 2005; Wang et al., 2009), videos (Niebles et al., 2008), survey data (Erosheva et al., 2007) or social networks data (Airoldi et al., 2007).

Since documents are frequently associated with other variables such as labels, tags or ratings, much interest has been placed on supervised topic models (Mcauliffe and Blei, 2008), which allow the use of that extra information to "guide" the topics discovery. By jointly learning the topics distributions and a prediction model, supervised topic models have been shown to outperform the separate use of their unsupervised analogues with an external regression or classification algorithm (Wang et al., 2009; Zhu et al., 2012).

Supervised topics models are then state-of-the-art approaches for predicting target variables associated with complex high-dimensional data, such as documents or images. Unfortunately, as we previously discussed, the size of modern datasets deem the use of a single annotator unrealistic and unfeasable for the majority of the real-world applications that involve some form of human labeling. For instance, the popular Reuters-21578 benchmark dataset was categorized by a group of personnel from Reuters Ltd and Carnegie Group, Inc. Similarly, the LabelMe[1] project asks volunteers to annotate images from a large collection using an online tool. Hence,

[1] http://labelme.csail.mit.edu

it is seldom the case where a single oracle labels an entire collection. Furthermore, through its social nature, the web also exploits the wisdom of crowds to annotate large collections of documents and images. By categorizing texts, tagging images or rating products, web users are generating large volumes of labeled content. However, as we saw in the previous chapters, when learning supervised models from crowds the quality of labels can vary significantly due to task subjectivity and annotator reliability (or bias) (Snow et al., 2008; Rodrigues et al., 2013a).

In this chapter, we propose a fully generative supervised topic model that is able to account for the different reliabilities of multiple annotators and correct their biases. The proposed model is capable of jointly modeling the words in documents as arising from a mixture of topics, the latent true labels as a result of the empirical distribution over topics of the documents, and the labels of the multiple annotators as noisy versions of that latent ground truth. This contrasts with the previous chapters, where we assumed the input vectors (or features) to be fixed. Hence, no model of the inputs was considered. Although one could treat the two problems separately, i.e. use a topic model to build a lower-dimensional representation of the data and apply a multiple-annotator model such as the ones developed in the previous chapters, this solution is suboptimal, since the information from the target variables is not being used to "guide" the topics discovery, which would allow the model to produce more discriminative topics. As we shall see in Section 5.5, approaching the two problems jointly gives significantly better results.

We propose two different models, one for classification and another for regression problems, thus covering a very wide range of possible practical application, as we demonstrate in Section 5.5. Since the majority of the tasks for which multiple annotators are used generally involve complex data such as text, images and video, by developing a multi-annotator supervised topic model we are contributing with a powerful tool for learning predictive models of complex high-dimensional data from crowds.

Given that the increasing sizes of modern datasets can pose a problem for obtaining human labels as well as for Bayesian inference, we propose efficient stochastic variational inference algorithms (Hoffman et al., 2013) that are able to scale to very large datasets. We empirically show, using both simulated and real multiple-annotator labels obtained from AMT for popular text and image collections, that the proposed models are able to outperform other state-of-the-art approaches in both classification and regression tasks. We further show the computational and predictive advantages of stochastic variational inference algorithms over their batch counterparts.

The remainder of this chapter is organized as follows: Section 5.2 provides a literature review on supervised topic models; Sections 5.3 and 5.4 describes the proposed models for classification and regression, respectively; in Section 5.5, we empirically evaluate the proposed models and finally, in Section 5.6, we conclude.

5.2 Supervised topic models

In this section, we review the literature on supervised topic models. But before we proceed, let us first review the simplest and, at the same time, the most popular unsupervised topic model in the literature: latent Dirichlet allocation (LDA), as this will allow us to better understand its supervised counterparts.

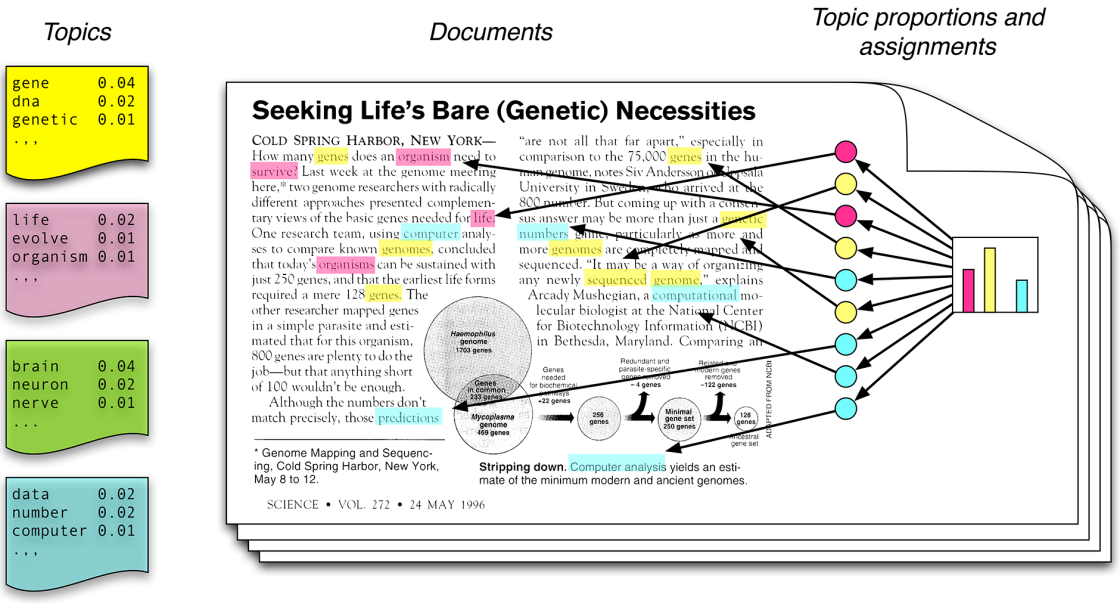


Figure 5.1: Intuition behind LDA (source: Blei (2012))

LDA is a generative model of the words in documents. The basic intuition behind LDA is that documents exhibit multiple topics, which are represented as probability distributions over words. The documents are then assumed to be mixtures of the corpus-wide topics, such that each individual word in a document is drawn from one of those topics. Figure 5.1 illustrates this intuition. In this example, we can see the topics on the left, as distributions over words, and an article, entitled "Seeking Life's Bare (Genetic) Necessities", being represented as a mixture of topics (the histogram at right). Each word in the article is then assigned a topic, represented by the colored circles.

Translating this intuition into a generative model leads to the generative process of LDA, which can be summarized as follows:

1. For each topic k
   (a) Draw topic's distribution over words $\boldsymbol{\beta}_k|\tau \sim \text{Dirichlet}(\boldsymbol{\beta}_k|\tau\mathbf{1}_V)$
2. For each document d
   (a) Draw topic proportions $\boldsymbol{\theta}^d|\alpha \sim \text{Dirichlet}(\boldsymbol{\theta}^d|\alpha\mathbf{1}_K)$
   (b) For the n^{th} word
      i. Draw topic assignment $z_n^d|\boldsymbol{\theta}^d \sim \text{Multinomial}(z_n^d|\boldsymbol{\theta}^d)$
      ii. Draw word $w_n^d|z_n^d, \boldsymbol{\beta}_{1:K} \sim \text{Multinomial}(w_n^d|\boldsymbol{\beta}_{z_n^d})$

where K denotes the number of topics and V is the length of the vocabulary. The topic proportions are drawn from a Dirichlet distribution parameterized by α, which controls the mean shape and sparsity of $\boldsymbol{\theta}^d$. High values of α (e.g., $\alpha > 1$) lead to smooth distributions, while small values ($\alpha \leq 1$) lead to sparse distributions. The same applies to the parameter τ. Figure 5.2 shows a graphical model representation of LDA.

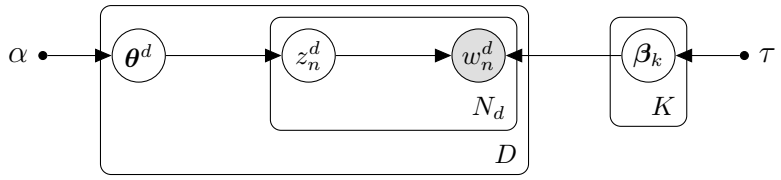


Figure 5.2: Graphical model representation of LDA.

Topic 247		Topic 5		Topic 43		Topic 56	
word	prob	word	prob	word	prob	word	prob
drugs	.069	red	.202	mind	.081	doctor	.074
drug	.060	blue	.099	thought	.066	dr.	.063
medicine	.027	green	.096	remember	.064	patient	.061
effects	.026	yellow	.073	memory	.037	hospital	.049
body	.023	white	.048	thinking	.030	care	.046
medicines	.019	color	.048	professor	.028	medical	.042
pain	.016	bright	.030	felt	.025	nurse	.031
person	.016	colors	.029	remembered	.022	patients	.029
marijuana	.014	orange	.027	thoughts	.020	doctors	.028
label	.012	brown	.027	forgotten	.020	health	.025
alcohol	.012	pink	.017	moment	.020	medicine	.017
dangerous	.011	look	.017	think	.019	nursing	.017
abuse	.009	black	.016	thing	.016	dental	.015
effect	.009	purple	.015	wonder	.014	nurses	.013
known	.008	cross	.011	forget	.012	physician	.012
pils	.008	colored	.009	recall	.012	hospitals	.011

Table 5.1: Example of four topics extracted from the TASA corpus in (Steyvers and Griffiths, 2007).

The goal with LDA is then to infer the posterior distribution over the latent structure, namely, the per-document topic proportions $\boldsymbol{\theta}^d$, the per-word topic assignments z_n^d and the per-topic distribution over words $\boldsymbol{\beta}_k$. This posterior is intractable to compute exactly. Hence, approximate Bayesian inference such as Gibbs sampling (Andrieu et al., 2003; Steyvers and Griffiths, 2007) and variational inference (Blei et al., 2003; Murphy, 2012) are typically used. Table 5.1 shows four examples of topics inferred from the Touchstone Applied Science Associates corpus (Zeno et al., 1995). The words are downwardly sorted by their probability under the topic, which means that the words that best represent each topic are in the top positions. Clearly, topics join the words semantically related. In the topic 247 are words related to drugs, in the topic 5, to colors, in the 43^{rd} topic, to mind and, in the topic 56, words relate to medical visits. Since each document is assigned to a distribution over topics, a document about color theory would have topic 5 as its main topic and a medical article would probably have the 56^{th} and 247^{th} topics as its most likely topics.

Latent Dirichlet allocation (LDA) soon proved to be a powerful tool for modeling documents (Blei et al., 2003) and images (Fei-Fei and Perona, 2005), by extracting their underlying topics. However, the need to model the relationship between docu-

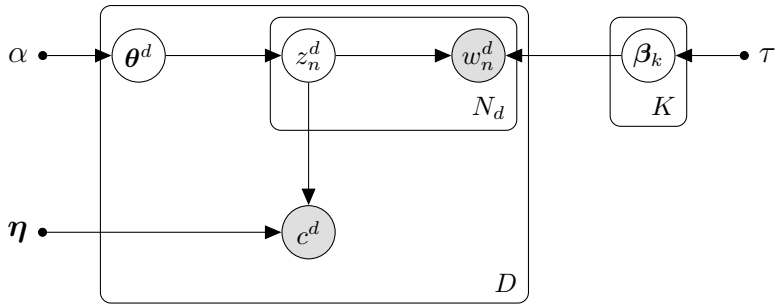


Figure 5.3: Graphical model representation of sLDA.

ments and labels quickly gave rise to many supervised variants of LDA. One of the first notable works was that of Mcauliffe and Blei (2008) in developing supervised LDA (sLDA). By extending LDA through the inclusion of a response variable that is linearly dependent on the mean topic-assignments of the words in a document, sLDA is able to jointly model the documents and their responses, in order to find the latent topics that will best predict the response variables for future unlabeled documents. Although initially developed for general continuous response variables, Wang et al. (2009) later extended sLDA to classification problems, by modeling the relationship between topic-assignments and labels with a softmax function. Letting c^d denote the class of the d^{th} document, the generative process of sLDA is as follows:

1. For each topic k
   (a) Draw topic's distribution over words $\boldsymbol{\beta}_k|\tau \sim \text{Dirichlet}(\boldsymbol{\beta}_k|\tau\mathbf{1}_V)$
2. For each document d
   (a) Draw topic proportions $\boldsymbol{\theta}^d|\alpha \sim \text{Dirichlet}(\boldsymbol{\theta}^d|\alpha\mathbf{1}_K)$
   (b) For the n^{th} word
      i. Draw topic assignment $z_n^d|\boldsymbol{\theta}^d \sim \text{Multinomial}(z_n^d|\boldsymbol{\theta}^d)$
      ii. Draw word $w_n^d|z_n^d, \boldsymbol{\beta}_{1:K} \sim \text{Multinomial}(w_n^d|\boldsymbol{\beta}_{z_n^d})$
   (c) Draw class label $c^d|\boldsymbol{z}^d, \boldsymbol{\eta} \sim \text{Multinomial}(c^d|\text{Softmax}(\bar{\boldsymbol{z}}^d, \boldsymbol{\eta}))$

where $\bar{\boldsymbol{z}}^d$ is the mean topic-assignment for document d, i.e. $\bar{\boldsymbol{z}}^d = \frac{1}{N_d}\sum_{n=1}^{N_d} z_n^d$, and $p(c^d|\bar{\boldsymbol{z}}^d, \boldsymbol{\eta})$ is a multi-class logistic regression model, such that

$$p(c^d|\bar{\boldsymbol{z}}^d, \boldsymbol{\eta}) = \frac{\exp(\boldsymbol{\eta}_{c^d}^{\mathrm{T}}\bar{\boldsymbol{z}}^d)}{\sum_{l=1}^{C}\exp(\boldsymbol{\eta}_l^{\mathrm{T}}\bar{\boldsymbol{z}}^d)}. \tag{5.1}$$

Notice that, while z_n^d is a nominal random variable represented using a 1-of-K encoding and therefore denoted by a non-bold letter, the average of all z_n^d in a document, $\bar{\boldsymbol{z}}^d = \frac{1}{N_d}\sum_{n=1}^{N_d} z_n^d$, becomes a vector, which is made clear by the use of a bold letter. The sLDA model then introduces a new set of parameters $\boldsymbol{\eta}$, the coefficients of the logistic regression model, which can be estimated using a variational Bayesian EM (VBEM) procedure (Wang et al., 2009). Figure 5.3 shows the graphical model corresponding to sLDA.

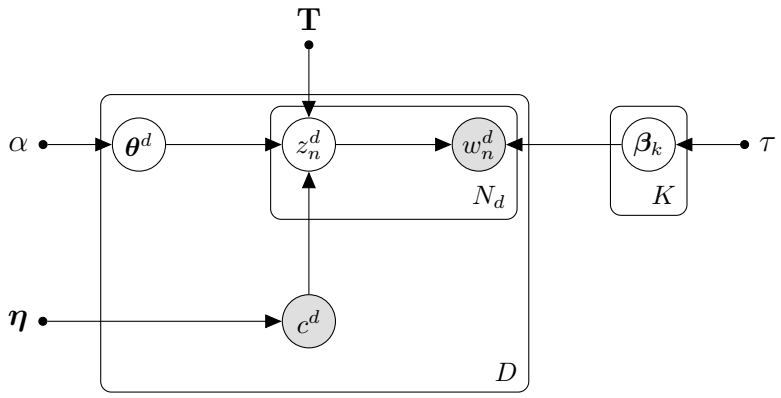


Figure 5.4: Graphical model representation of DiscLDA.

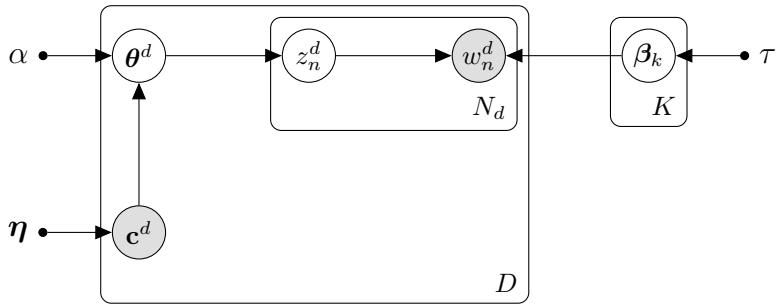


Figure 5.5: Graphical model representation of Labeled-LDA.

From a classification perspective, there are several ways in which document classes can be included in LDA. The most natural one in this setting is probably the sLDA approach, since the classes are directly dependent on the empirical topic mixture distributions. This approach is coherent with the generative perspective of LDA but, nevertheless, several discriminative alternatives also exist. For example, DiscLDA (Lacoste-Julien et al., 2009) introduces a class-dependent linear transformation on the topic mixture proportions $\boldsymbol{\theta}^d$. Hence, the per-word topic assignment z_n^d are now drawn from the linearly transformed mixture proportions, i.e. $z_n^d \sim \text{Multinomial}(z_n^d|\mathbf{T}_{c^d}\boldsymbol{\theta}^d)$. The class-specific transformation matrices $\mathbf{T}_{c^d}$ are then able to reposition the vectors $\boldsymbol{\theta}^d$ such that documents with the same class labels have similar topics mixture proportions. Figure 5.4 shows a graphical model representation. In the case of DiscLDA, the parameters $\mathbf{T}_{1:C}$ are estimated by maximizing the conditional likelihood of response variables (see Lacoste-Julien et al. (2009)).

An alternative way of including classes in LDA for supervision is the one proposed by Ramage et al. (2009) in their Labeled-LDA model. Labeled-LDA is a variant of LDA that incorporates supervision by constraining the topic model to assign to a document only the topics that correspond to its label set. This is achieved by introducing the per-document matrix $\mathbf{L}^d$, with values:

$$\mathbf{L}_{i,j}^d = \begin{cases} 1, & \text{if } c_i^d = j. \\ 0, & \text{otherwise.} \end{cases} \tag{5.2}$$

The documents' mixture proportions $\boldsymbol{\theta}^d$ are then drawn from a Dirichlet distribution with parameters $\mathbf{L}^d(\alpha \mathbf{1}_K)$, thus forcing the mixture proportions to contain only the topics corresponding to the classes that the document belongs to. Notice that, while this has the advantage of allowing multiple labels per document, it is restrictive in the sense that the number of topics needs to be the same as the number of possible labels. Figure 5.5 shows the graphical model representation of Labeled-LDA, thus evidencing the differences to DiscLDA. Notice that, since Labeled-LDA allows multiple labels per document, a bold letter $\mathbf{c}^d$ is used to represent the classes. The vector $\mathbf{c}^d$ is then a C-dimensional binary vector (where C denotes the number of classes) indicating which classes the d^{th} document belongs to.

From a regression perspective, other than sLDA, the most relevant approaches are the Dirichlet-multimonial regression (Mimno and McCallum, 2008) and the inverse regression topic models (Rabinovich and Blei, 2014). The Dirichlet-multimonial regression (DMR) topic model (Mimno and McCallum, 2008) includes a log-linear prior on the document's mixture proportions that is a function of a set of arbitrary features, such as author, date, publication venue or references in scientific articles. The inferred Dirichlet-multinomial distribution can then be used to make predictions about the values of theses features. The inverse regression topic model (IRTM) (Rabinovich and Blei, 2014) is a mixed-membership extension of the multinomial inverse regression (MNIR) model proposed by Taddy (2013) that exploits the topical structure of text corpora to improve its predictions and facilitate exploratory data analysis. However, this results in a rather complex and inefficient inference procedure. Furthermore, making predictions in the IRTM is not trivial. For example, MAP estimates of target variables will be in a different scale than the original document's metadata. Hence, the authors propose the use of a linear model to regress metadata values onto their MAP predictions.

The approaches discussed so far rely on likelihood-based estimation procedures. The work of Zhu et al. (2012) contrasts with these approaches by proposing MedLDA, a supervised topic model that utilizes the max-margin principle for estimation. Despite its margin-based advantages, MedLDA looses the probabilistic interpretation of the document classes given the topic mixture distributions. On the contrary, this chapter proposes two fully generative probabilistic models of the labels of multiple annotators and the words in the documents.

5.3 Classification model

In this section, we develop a multi-annotator supervised topic model for classification problems. The model for regression settings will be presented in Section 5.4. We start by deriving a (*batch*) variational inference algorithm for approximating the posterior distribution over the latent variables and an algorithm to estimate the model parameters. We then develop a stochastic variational inference algorithm that gives the model the capability of handling large collections of documents. Finally, we show how to use the learned model to classify new documents.

5.3.1 Proposed model

Let $\mathcal{D} = \{\mathbf{w}^d, \mathbf{y}^d\}_{d=1}^D$ be an annotated corpus of size D, where each document $\mathbf{w}^d = \{w_n^d\}_{n=1}^{N_d}$ is given a set of labels $\mathbf{y}^d = \{y_r^d\}_{r=1}^R$ from R distinct annotators. We can take

advantage of the inherent topical structure of documents and model their words as arising from a mixture of topics, each being defined as a distribution over the words in a vocabulary, as in LDA. In LDA, the n^{th} word, w_n^d, in a document d is provided a discrete topic-assignment z_n^d, which is drawn from the documents' distribution over topics $\boldsymbol{\theta}^d$. This allows us to build lower-dimensional representations of documents, which we can explore to build classification models by assigning coefficients $\boldsymbol{\eta}$ to the mean topic-assignment of the words in the document, $\bar{\mathbf{z}}^d$, and applying a softmax function in order to obtain a distribution over classes.

Unfortunately, a direct mapping between document classes and the labels provided by the different annotators in a multiple-annotator setting would correspond to assuming that they are all equally reliable, an assumption that is violated in practice, as previous works clearly demonstrate (e.g. Snow et al. (2008); Rodrigues et al. (2013a)). Hence, we assume the existence of a latent ground truth class, and model the labels from the different annotators using a noise model that states that, given a true class c, each annotator r provides the label l with some probability $\pi_{c,l}^r$. Hence, by modeling the matrix $\boldsymbol{\Pi}^r = \{\boldsymbol{\pi}_c^r\}_{c=1}^C$, where C denotes the number of classes, we are in fact modeling a per-annotator confusion matrix, which allows us to account for their different levels of expertise and correct their potential biases.

The generative process of the proposed model for classification problems can then be summarized as follows:

1. For each annotator r
   (a) For each class c
      i. Draw annotator reliability parameter $\boldsymbol{\pi}_c^r|\omega \sim \text{Dirichlet}(\boldsymbol{\pi}_c^r|\omega\mathbf{1}_C)$
2. For each topic k
   (a) Draw topic's distribution over words $\boldsymbol{\beta}_k|\tau \sim \text{Dirichlet}(\boldsymbol{\beta}_k|\tau\mathbf{1}_V)$
3. For each document d
   (a) Draw topic proportions $\boldsymbol{\theta}^d|\alpha \sim \text{Dirichlet}(\boldsymbol{\theta}^d|\alpha\mathbf{1}_K)$
   (b) For the n^{th} word
      i. Draw topic assignment $z_n^d|\boldsymbol{\theta}^d \sim \text{Multinomial}(z_n^d|\boldsymbol{\theta}^d)$
      ii. Draw word $w_n^d|z_n^d, \boldsymbol{\beta}_{1:K} \sim \text{Multinomial}(w_n^d|\boldsymbol{\beta}_{z_n^d})$
   (c) Draw latent (true) class $c^d|\mathbf{z}^d, \boldsymbol{\eta} \sim \text{Multinomial}(c^d|\text{Softmax}(\bar{\boldsymbol{z}}^d, \boldsymbol{\eta}))$
   (d) For each annotator r
      i. Draw annotator's answer $y^{d,r}|c^d, \boldsymbol{\Pi}^r \sim \text{Multinomial}(y^{d,r}|\boldsymbol{\pi}_{c^d}^r)$

where $\bar{\boldsymbol{z}}^d$ is the mean topic-assignment for document d, i.e. $\bar{\boldsymbol{z}}^d = \frac{1}{N_d}\sum_{n=1}^{N_d} z_n^d$, and $p(c^d|\bar{\boldsymbol{z}}^d, \boldsymbol{\eta})$ is a multi-class logistic regression model, such that

$$p(c^d|\bar{\boldsymbol{z}}^d, \boldsymbol{\eta}) = \frac{\exp(\boldsymbol{\eta}_{c^d}^{\mathrm{T}}\bar{\boldsymbol{z}}^d)}{\sum_{l=1}^{C}\exp(\boldsymbol{\eta}_l^{\mathrm{T}}\bar{\boldsymbol{z}}^d)}. \tag{5.3}$$

Figure 5.6 shows a graphical model representation of the proposed model, where K denotes the number of topics, R denotes the number of annotators, and N_d is the number of words in the d^{th} document. Notice that we included a Dirichlet prior

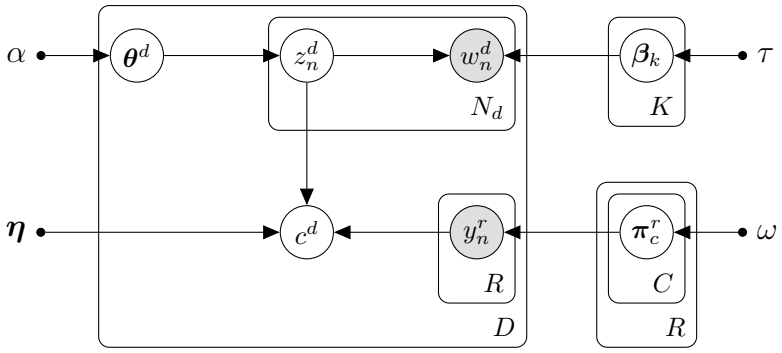


Figure 5.6: Graphical representation of the proposed model for classification.

over the topics $\boldsymbol{\beta}_k$ to produce a smooth posterior and control sparsity. Similarly, instead of computing maximum likelihood or MAP estimates for the annotators reliability parameters $\boldsymbol{\pi}_c^r$, we place a Dirichlet prior over these variables and perform approximate Bayesian inference. This contrasts with previous works on learning from crowds (e.g. Raykar et al. (2010); Yan et al. (2010)).

5.3.2 Approximate inference

Given a dataset $\mathcal{D} = \{\mathbf{W}, \mathbf{Y}\}$, where $\mathbf{W} = \{\mathbf{w}^d\}_{d=1}^D$ and $\mathbf{Y} = \{\mathbf{y}^d\}_{d=1}^D$, the goal of inference is to compute the posterior distributions of: the per-document topic proportions $\boldsymbol{\theta}^d$, the per-word topic assignments z_n^d, the per-topic distribution over words $\boldsymbol{\beta}_k$, the per-document latent true class c^d, and the per-annotator confusion parameters $\boldsymbol{\Pi}^r$. As with LDA, computing the exact posterior distribution of the latent variables is computationally intractable. Hence, we employ mean-field variational inference to perform approximate Bayesian inference.

According to the graphical model (and the generative process), the joint distribution of the proposed model factorizes as

$$p(\boldsymbol{\theta}_{1:D}, \mathbf{z}_{1:D}, \boldsymbol{c}, \boldsymbol{\beta}_{1:K}, \boldsymbol{\Pi}_{1:R}, \mathbf{W}, \mathbf{Y}|\Theta) = \left(\prod_{r=1}^{R}\prod_{c=1}^{C} p(\boldsymbol{\pi}_c^r|\omega)\right)\left(\prod_{i=1}^{K} p(\boldsymbol{\beta}_i|\tau)\right)$$
$$\times \prod_{d=1}^{D} p(\boldsymbol{\theta}^d|\alpha)\left(\prod_{n=1}^{N_d} p(z_n^d|\boldsymbol{\theta}^d)\, p(w_n^d|z_n^d, \boldsymbol{\beta}_{1:K})\right) p(c^d|\mathbf{z}^d, \boldsymbol{\eta}) \prod_{r=1}^{R} p(y^{d,r}|c^d, \boldsymbol{\Pi}^r),$$

where $\Theta = \{\alpha, \tau, \omega, \boldsymbol{\eta}\}$ denotes the model parameters.

Variational inference methods seek to minimize the KL divergence between the variational and the true posterior distribution. We assume a fully-factorized (mean-field) variational distribution of the form

$$q(\boldsymbol{\theta}_{1:D}, \mathbf{z}_{1:D}, \boldsymbol{c}, \boldsymbol{\beta}_{1:K}, \boldsymbol{\Pi}_{1:R}) = \left(\prod_{r=1}^{R}\prod_{c=1}^{C} q(\boldsymbol{\pi}_c^r|\boldsymbol{\xi}_c^r)\right)\left(\prod_{i=1}^{K} q(\boldsymbol{\beta}_i|\boldsymbol{\zeta}_i)\right)$$
$$\times \prod_{d=1}^{D} q(\boldsymbol{\theta}^d|\boldsymbol{\gamma}^d)\left(\prod_{n=1}^{N_d} q(z_n^d|\boldsymbol{\phi}_n^d)\right) q(c^d|\boldsymbol{\lambda}^d).$$

The values $\boldsymbol{\Xi}_{1:R}$, $\boldsymbol{\zeta}_{1:K}$, $\boldsymbol{\gamma}_{1:D}$, $\boldsymbol{\lambda}$ and $\boldsymbol{\Phi}_{1:D}$ are the variational parameters, where we introduced the notation $\boldsymbol{\Xi}^r = \{\boldsymbol{\xi}_c^r\}_{c=1}^C$ and $\boldsymbol{\Phi}^d = \{\boldsymbol{\phi}_n^d\}_{n=1}^{N_d}$. Table 5.2 shows the correspondence between variational parameters and the original parameters.

Following Jordan et al. (1999) (see Section 2.2.2 for details), the KL minimization can be equivalently formulated as maximizing the following lower bound on the log marginal likelihood

$$\begin{aligned}
\log\, p(\mathcal{D}|\Theta) &= \log \int \sum_{\mathbf{z},\mathbf{c}} q(\boldsymbol{\theta}_{1:D}, \mathbf{z}_{1:D}, \boldsymbol{c}, \boldsymbol{\beta}_{1:K}, \boldsymbol{\Pi}_{1:R}) \\
&\quad \times \frac{p(\boldsymbol{\theta}_{1:D}, \mathbf{z}_{1:D}, \boldsymbol{c}, \boldsymbol{\beta}_{1:K}, \boldsymbol{\Pi}_{1:R}, \mathbf{W}, \mathbf{Y}|\Theta)}{q(\boldsymbol{\theta}_{1:D}, \mathbf{z}_{1:D}, \boldsymbol{c}, \boldsymbol{\beta}_{1:K}, \boldsymbol{\Pi}_{1:R})}\, d\boldsymbol{\theta}_{1:D}\, d\boldsymbol{\beta}_{1:K}\, d\boldsymbol{\Pi}_{1:R} \\
&\geqslant \mathbb{E}_q[\log p(\boldsymbol{\theta}_{1:D}, \mathbf{z}_{1:D}, \boldsymbol{c}, \boldsymbol{\beta}_{1:K}, \boldsymbol{\Pi}_{1:R}, \mathbf{W}, \mathbf{Y}|\Theta)] \\
&\quad - \mathbb{E}_q[\log q(\boldsymbol{\theta}_{1:D}, \mathbf{z}_{1:D}, \boldsymbol{c}, \boldsymbol{\beta}_{1:K}, \boldsymbol{\Pi}_{1:R})] \\
&= \mathcal{L}(\boldsymbol{\gamma}_{1:D}, \boldsymbol{\Phi}_{1:D}, \boldsymbol{\lambda}, \boldsymbol{\zeta}_{1:K}, \boldsymbol{\Xi}_{1:R}|\Theta),
\end{aligned} \tag{5.4}$$

which we maximize using coordinate ascent. Exploiting the factorization of the joint and the variational distributions, we can write the evidence lower bound $\mathcal{L}$ as

$$\begin{aligned}
&\mathcal{L}(\boldsymbol{\gamma}_{1:D}, \boldsymbol{\Phi}_{1:D}, \boldsymbol{\lambda}, \boldsymbol{\zeta}_{1:K}, \boldsymbol{\Xi}_{1:R}|\Theta) \\
&\quad = \mathbb{E}_q[\log p(\boldsymbol{\theta}_{1:D}, \mathbf{z}_{1:D}, \boldsymbol{c}, \boldsymbol{\beta}_{1:K}, \boldsymbol{\Pi}_{1:R}, \mathbf{W}, \mathbf{Y}|\Theta)] \\
&\quad \underbrace{-\mathbb{E}_q[\log q(\boldsymbol{\theta}_{1:D}, \mathbf{z}_{1:D}, \boldsymbol{c}, \boldsymbol{\beta}_{1:K}, \boldsymbol{\Pi}_{1:R})]}_{\mathcal{H}(q)} \\
&\quad = \sum_{r=1}^{R}\sum_{c=1}^{C} \mathbb{E}_q[\log p(\boldsymbol{\pi}_c^r|\omega)] + \sum_{i=1}^{K} \mathbb{E}_q[\log p(\boldsymbol{\beta}_i|\tau)] \\
&\quad + \sum_{d=1}^{D} \Bigg(\mathbb{E}_q[\log p(\boldsymbol{\theta}^d|\alpha)] + \sum_{n=1}^{N^d} \mathbb{E}_q[\log p(z_n^d|\boldsymbol{\theta}^d)] + \sum_{n=1}^{N^d} \mathbb{E}_q[\log p(w_n^d|z_n^d, \boldsymbol{\beta}_{1:K})] \\
&\quad + \mathbb{E}_q[\log p(c^d|\bar{\mathbf{z}}^d, \boldsymbol{\eta})] + \sum_{r=1}^{R} \mathbb{E}_q[\log p(y^{d,r}|c^d, \boldsymbol{\Pi}^r)] \Bigg) + \mathcal{H}(q),
\end{aligned} \tag{5.5}$$

where the entropy $\mathcal{H}(q)$ of the variational distribution is given by

$$\begin{aligned}
\mathcal{H}(q) = &- \sum_{r=1}^{R}\sum_{c=1}^{C} \mathbb{E}_q[\log q(\boldsymbol{\pi}_c^r|\boldsymbol{\xi}_c^r)] - \sum_{i=1}^{K} \mathbb{E}_q[\log q(\boldsymbol{\beta}_i|\boldsymbol{\zeta}_i)] \\
&- \sum_{d=1}^{D} \Bigg(\mathbb{E}_q[\log q(\boldsymbol{\theta}^d|\boldsymbol{\gamma}^d)] - \sum_{n=1}^{N^d} \mathbb{E}_q[\log q(z_n^d|\boldsymbol{\phi}_n^d)] - \mathbb{E}_q[\log q(c^d|\boldsymbol{\lambda}^d)] \Bigg).
\end{aligned} \tag{5.6}$$

Please refer to Appendix C.2 for the details on how to compute each of these expectations individually. The fully-expanded expression for the evidence lower bound $\mathcal{L}$ is given in (C.10).

Optimizing $\mathcal{L}$ w.r.t. γ and ζ, by taking derivatives and setting them to zero,

Variational param.	Original param.	Description
$\boldsymbol{\Xi}^r = \{\boldsymbol{\xi}_c^r\}_{c=1}^C$	$\boldsymbol{\Pi}^r = \{\boldsymbol{\pi}_c^r\}_{c=1}^C$	per-annotator confusion parameters
$\boldsymbol{\zeta}_{1:K} = \{\boldsymbol{\zeta}_k\}_{k=1}^K$	$\boldsymbol{\beta}_{1:K} = \{\boldsymbol{\beta}_k\}_{k=1}^K$	per-topic distribution over words
$\boldsymbol{\gamma}_{1:D} = \{\boldsymbol{\gamma}^d\}_{d=1}^D$	$\boldsymbol{\theta}_{1:D} = \{\boldsymbol{\theta}^d\}_{d=1}^D$	per-document topic proportions
$\boldsymbol{\lambda}_{1:D} = \{\boldsymbol{\lambda}^d\}_{d=1}^D$	$\mathbf{c} = \{c^d\}_{d=1}^D$	per-document latent true class
$\boldsymbol{\Phi}^d = \{\boldsymbol{\phi}_n^d\}_{n=1}^N$	$\mathbf{z}^d = \{z_n^d\}_{n=1}^N$	per-word topic assignments

Table 5.2: Correspondence between variational parameters and the original parameters.

gives the same coordinate ascent updates as in (Blei et al., 2003), which are

$$\gamma_i^d = \alpha + \sum_{n=1}^{N_d} \phi_{n,i}^d \tag{5.7}$$

$$\zeta_{i,j} = \tau + \sum_{d=1}^{D}\sum_{n=1}^{N_d} w_{n,j}^d \phi_{n,i}^d. \tag{5.8}$$

The variational Dirichlet parameters ξ can be optimized by collecting only the terms in $\mathcal{L}$ (please refer to Eq. C.10) that contain ξ

$$\begin{aligned}\mathcal{L}_{[\boldsymbol{\xi}]} = &\sum_{r=1}^{R}\sum_{c=1}^{C}\sum_{l=1}^{C} \mathbb{E}_q[\log \pi_{c,l}^r]\left(\omega + \sum_{d=1}^{D_r} \lambda_c^d y_l^{d,r} - \xi_{c,l}^r\right) \\ &- \sum_{r=1}^{R}\sum_{c=1}^{C} \log\Gamma\left(\sum_{t=1}^{C} \xi_{c,t}^r\right) + \sum_{r=1}^{R}\sum_{c=1}^{C}\sum_{l=1}^{C} \log\Gamma(\xi_{c,l}^r),\end{aligned}$$

where D_r denotes the documents labeled by the r^{th} annotator, $\mathbb{E}_q[\log \pi_{c,l}^r] = \Psi(\xi_{c,l}^r) - \Psi(\sum_{t=1}^{C} \xi_{c,t}^r)$, and $\Gamma(\cdot)$ and $\Psi(\cdot)$ are the gamma and digamma functions, respectively. Taking derivatives of $\mathcal{L}_{[\boldsymbol{\xi}]}$ w.r.t. $\xi_{c,l}^r$ and setting them to zero, yields the following update

$$\xi_{c,l}^r = \omega + \sum_{d=1}^{D_r} \lambda_c^d y_l^{d,r}. \tag{5.9}$$

Similarly, the coordinate ascent updates for the documents distribution over classes λ can be found by considering the terms in $\mathcal{L}$ that contain λ

$$\mathcal{L}_{[\boldsymbol{\lambda}]} = \sum_{d=1}^{D}\sum_{l=1}^{C} \lambda_l^d \boldsymbol{\eta}_l^{\mathrm{T}} \bar{\boldsymbol{\phi}}^d - \sum_{l=1}^{C} \lambda_l^d \log \lambda_l^d + \sum_{d=1}^{D}\sum_{r=1}^{R}\sum_{l=1}^{C}\sum_{c=1}^{C} \lambda_l^d \, y_c^{d,r} \, \mathbb{E}_q[\log \pi_{l,c}^r],$$

where $\bar{\boldsymbol{\phi}}^d = \frac{1}{N_d}\sum_{n=1}^{N_d} \boldsymbol{\phi}_n^d$. Adding the necessary Lagrange multipliers to ensure that $\sum_{l=1}^{C} \lambda_l^d = 1$ and setting the derivatives w.r.t. λ_l^d to zero gives the following update

$$\lambda_l^d \propto \exp\left(\boldsymbol{\eta}_l^{\mathrm{T}} \bar{\boldsymbol{\phi}}^d + \sum_{r=1}^{R}\sum_{c=1}^{C} y_c^{d,r} \, \mathbb{E}_q[\log \pi_{l,c}^r]\right). \tag{5.10}$$

Observe how the variational distribution over the true classes results from a combination between the dot product of the inferred mean topic assignment $\bar{\boldsymbol{\phi}}^d$ with the coefficients $\boldsymbol{\eta}$ and the labels $\mathbf{y}^d$ from the multiple annotators "weighted" by their expected log probability $\mathbb{E}_q[\log \pi^r_{l,c}]$.

The main difficulty of applying standard variational inference methods to the proposed model is the non-conjugacy between the distribution of the mean topic-assignment $\bar{\mathbf{z}}^d$ and the softmax. Namely, in the expectation

$$\mathbb{E}_q[\log p(c^d|\bar{\mathbf{z}}^d, \boldsymbol{\eta})] = \mathbb{E}_q\left[\log \frac{\exp(\boldsymbol{\eta}_{c^d}^{\mathrm{T}}\bar{\mathbf{z}}^d)}{\sum_{l=1}^C \exp(\boldsymbol{\eta}_l^{\mathrm{T}}\bar{\mathbf{z}}^d)}\right] = \mathbb{E}_q[\boldsymbol{\eta}_{c^d}^{\mathrm{T}}\bar{\mathbf{z}}^d] - \mathbb{E}_q\left[\log \sum_{l=1}^C \exp(\boldsymbol{\eta}_l^{\mathrm{T}}\bar{\mathbf{z}}^d)\right],$$

the second term is intractable to compute. We can make progress by applying Jensen's inequality, which states that $\log(\mathbb{E}[x]) \geqslant \mathbb{E}[\log(x)]$, to bound it as follows

$$\begin{aligned}
-\mathbb{E}_q\left[\log \sum_{l=1}^C \exp(\boldsymbol{\eta}_l^{\mathrm{T}}\bar{\mathbf{z}}^d)\right] &\geqslant -\log \sum_{l=1}^C \mathbb{E}_q[\exp(\boldsymbol{\eta}_l^{\mathrm{T}}\bar{\mathbf{z}}^d)] \\
&= -\log \sum_{l=1}^C \prod_{j=1}^{N_d} \left(\boldsymbol{\phi}_j^d\right)^{\mathrm{T}} \exp\left(\boldsymbol{\eta}_l \frac{1}{N_d}\right) \\
&= -\log(\mathbf{a}^{\mathrm{T}}\boldsymbol{\phi}_n^d), \qquad (5.11)
\end{aligned}$$

where $\mathbf{a} \triangleq \sum_{l=1}^C \exp(\frac{\boldsymbol{\eta}_l}{N_d}) \prod_{j=1, j\neq n}^{N_d} \left(\boldsymbol{\phi}_j^d\right)^{\mathrm{T}} \exp\left(\frac{\boldsymbol{\eta}_l}{N_d}\right)$, which is constant w.r.t. $\boldsymbol{\phi}_n^d$. This local variational bound can be made tight by noticing that $\log(x) \leqslant \epsilon^{-1}x + \log(\epsilon) - 1, \forall x > 0, \epsilon > 0$, where the equality holds if and only if $x = \epsilon$. Hence, given the current parameter estimates $(\boldsymbol{\phi}_n^d)^{old}$, if we set $x = \mathbf{a}^{\mathrm{T}}\boldsymbol{\phi}_n^d$ and $\epsilon = \mathbf{a}^{\mathrm{T}}(\boldsymbol{\phi}_n^d)^{old}$ then, for an individual parameter $\boldsymbol{\phi}_n^d$, we have that

$$-\mathbb{E}_q\left[\log \sum_{l=1}^C \exp(\boldsymbol{\eta}_l^{\mathrm{T}}\bar{\mathbf{z}}^d)\right] \geqslant -(\mathbf{a}^{\mathrm{T}}(\boldsymbol{\phi}_n^d)^{old})^{-1}(\mathbf{a}^{\mathrm{T}}\boldsymbol{\phi}_n^d) - \log(\mathbf{a}^{\mathrm{T}}(\boldsymbol{\phi}_n^d)^{old}) + 1.$$

Using this local bound to approximate the expectation of the log-sum-exp term, and taking derivatives of the evidence lower bound w.r.t. $\phi_{n,i}^d$ with the constraint that $\sum_{i=1}^K \phi_{n,i}^d = 1$, yields the following fix-point update

$$\begin{aligned}
\phi_{n,i}^d \propto \exp\Bigg(&\Psi(\gamma_i^d) + \sum_{j=1}^V w_{n,j}^d \left(\Psi(\zeta_{i,j}) - \Psi\left(\sum_{k=1}^V \zeta_{i,k}\right)\right) \\
&+ \frac{\sum_{l=1}^C \lambda_l^d \eta_{l,i}}{N_d} - (\mathbf{a}^{\mathrm{T}}(\boldsymbol{\phi}_n^d)^{old})^{-1} a_i\Bigg). \qquad (5.12)
\end{aligned}$$

where V denotes the size of the vocabulary. Notice how the per-word variational distribution over topics $\boldsymbol{\phi}_n^d$ depends on the variational distribution over the true class label $\boldsymbol{\lambda}^d$.

The variational inference algorithm iterates between equations 5.7-5.12 until the evidence lower bound, Eq. 5.5, converges. See Appendix C.2 for additional details on the derivation of this algorithm.

5.3.3 Parameter estimation

The model parameters are $\Theta = \{\alpha, \tau, \omega, \boldsymbol{\eta}\}$. For the sake of simplicity we assume the parameters α, τ and ω of the Dirichlet priors to be fixed, and only estimate the coefficients $\boldsymbol{\eta}$ using a variational Bayesian EM (VBEM) algorithm. Therefore, in the E-step we use the variational inference algorithm from section 5.3.2 to estimate the posterior distribution of the latent variables, and in the M-step we find maximum likelihood estimates of $\boldsymbol{\eta}$ by maximizing the evidence lower bound $\mathcal{L}$. Unfortunately, taking derivatives of $\mathcal{L}$ w.r.t. $\boldsymbol{\eta}$ does not yield a closed-form solution, hence we use a numerical method, namely L-BFGS (Nocedal and Wright, 2006), to find an optimum. The objective function and gradients are given by

$$\mathcal{L}_{[\eta]} = \sum_{d=1}^{D} \left(\sum_{l=1}^{C} \lambda_l^d \boldsymbol{\eta}_l^{\mathrm{T}} \bar{\boldsymbol{\phi}}^d - \log \sum_{l=1}^{C} b_l^d \right)$$

$$\nabla_{\eta_{l,i}} = \sum_{d=1}^{D} \left(\lambda_{l,i}^d \bar{\phi}_i^d - \frac{b_l^d}{\sum_{t=1}^{C} b_t^d} \sum_{n=1}^{N_d} \frac{\frac{1}{N_d} \phi_{n,i}^d \exp(\frac{1}{N_d} \eta_{l,i})}{\sum_{j=1}^{K} \phi_{n,j}^d \exp(\frac{1}{N_d} \eta_{l,j})} \right),$$

where, for convenience, we defined

$$b_l^d \triangleq \prod_{n=1}^{N_d} \Big(\sum_{i=1}^{K} \phi_{n,i}^d \exp\Big(\frac{1}{N_d} \eta_{l,i}\Big)\Big). \tag{5.13}$$

5.3.4 Stochastic variational inference

In section 5.3.2 we developed a batch coordinate ascent algorithm for performing variational inference in the proposed model. This algorithm iterates between analyzing every document in the corpus to infer the local hidden structure, and estimating the global hidden variables. However, this can be inefficient for large datasets, since it requires a full pass through the data at each iteration before updating the global variables, $\boldsymbol{\Xi}_{1:R}$ and $\boldsymbol{\zeta}_{1:K}$. In this section we develop a stochastic variational inference algorithm (Hoffman et al., 2013), which follows noisy estimates of the gradients of the evidence lower bound $\mathcal{L}$.

Based on the theory of stochastic optimization (Robbins and Monro, 1951), we can find unbiased estimates of the gradients by subsampling a document (or a mini-batch of documents) from the corpus, and using it to compute the gradients as if that document was observed D times. Hence, given an uniformly sampled document d, we use the current posterior distributions of the global latent variables, $\boldsymbol{\beta}$ and $\boldsymbol{\Pi}_{1:R}$, and the current coefficient estimates $\boldsymbol{\eta}_{1:K}$, to compute the posterior distribution over the local hidden variables $\boldsymbol{\theta}^d$, $\mathbf{z}^d$ and c^d using (5.7), (5.12) and (5.10) respectively. These posteriors are then used to update the global variational parameters, $\boldsymbol{\zeta}_{1:K}$ and $\boldsymbol{\Xi}_{1:R}$ by taking a step of size ρ_t in the direction of the noisy estimates of the natural gradients.

Algorithm 1 describes a stochastic variational inference algorithm for the proposed model. Given an appropriate schedule for the learning rates $\{\rho_t\}$, such that $\sum_t \rho_t$ and $\sum_t \rho_t^2 < \infty$, the stochastic optimization algorithm is guaranteed to converge to a local maximum of the evidence lower bound (Robbins and Monro, 1951).

Algorithm 1 Stochastic variational inference

Initialize $\boldsymbol{\gamma}_{1:D}^{(0)}$, $\boldsymbol{\phi}_{1:D}^{(0)}$, $\boldsymbol{\lambda}_{1:D}^{(0)}$, $\boldsymbol{\zeta}_{1:K}^{(0)}$, $\boldsymbol{\Xi}_{1:R}^{(0)}$, $t = 0$
repeat
  Set t = t + 1
  Sample a document $\mathbf{w}^d$ uniformly from the corpus
  repeat
    Compute $\boldsymbol{\phi}_n^d$ using (5.12), for $n \in \{1..N_d\}$
    Compute $\boldsymbol{\gamma}^d$ using (5.7)
    Compute $\boldsymbol{\lambda}^d$ using (5.10)
  until local parameters $\boldsymbol{\phi}_n^d$, $\boldsymbol{\gamma}^d$ and $\boldsymbol{\lambda}^d$ converge
  Compute step-size $\rho_t = (t + delay)^{-\kappa}$
  Update topics variational parameters

$$\zeta_{i,j}{}^{(t)} = (1 - \rho_t)\zeta_{i,j}^{(t-1)} + \rho_t\left(\tau + D\sum_{n=1}^{N_d} w_{n,j}^d \phi_{n,i}^d\right)$$

  Update annotators confusion parameters

$$\xi_{c,l}^r{}^{(t)} = (1 - \rho_t)\xi_{c,l}^r{}^{(t-1)} + \rho_t\big(\omega + D\,\lambda_c^d\, y_l^{d,r}\big)$$

until global convergence criterion is met

5.3.5 Document classification

In order to make predictions for a new (unlabeled) document d, we start by computing the approximate posterior distribution over the latent variables $\boldsymbol{\theta}^d$ and $\mathbf{z}^d$. This can be achieved by dropping the terms that involve y, c and π from the model's joint distribution (since, at prediction time, the multi-annotator labels are no longer observed) and averaging over the estimated topics distributions. Letting the topics distribution over words inferred during training be $q(\boldsymbol{\beta}_{1:K}|\boldsymbol{\zeta}_{1:K}) = \prod_{i=1}^K q(\boldsymbol{\beta}_i|\boldsymbol{\zeta}_i)$, the joint distribution for a single document is now simply given by

$$p(\boldsymbol{\theta}^d, \mathbf{z}^d) = \int q(\boldsymbol{\beta}_{1:K}|\boldsymbol{\zeta}_{1:K})\, p(\boldsymbol{\theta}^d|\alpha) \prod_{n=1}^{N_d} p(z_n^d|\boldsymbol{\theta}^d)\, p(w_n^d|z_n^d, \boldsymbol{\beta}_{1:K})\, d\boldsymbol{\beta}_{1:K}.$$

Deriving a mean-field variational inference algorithm for computing the posterior over $q(\theta^d, \mathbf{z}^d) = q(\theta^d|\gamma^d)\prod_{n=1}^{N_d} q(z_n^d|\phi_n^d)$ results in the same fixed-point updates as in LDA (Blei et al., 2003) for γ_i^d (Eq. 5.7) and $\phi_{n,i}^d$

$$\phi_{n,i}^d \propto \exp\left(\Psi(\gamma_i) + \sum_{j=1}^V w_{n,j}^d\left(\Psi(\zeta_{i,j}) - \Psi\left(\sum_{k=1}^V \zeta_{i,k}\right)\right)\right). \tag{5.14}$$

Using the inferred posteriors and the coefficients $\boldsymbol{\eta}$ estimated during training, we can make predictions as follows

$$c_*^d = \arg\max_c \eta_c^{\mathrm{T}} \bar{\boldsymbol{\phi}}^d. \tag{5.15}$$

This is equivalent to making predictions in the classification version of sLDA (Wang et al., 2009).

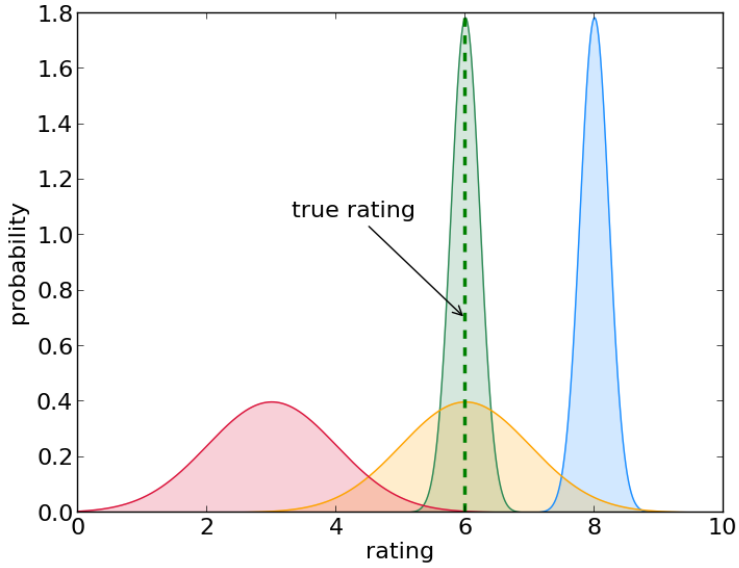


Figure 5.7: Example of 4 different annotators (represented by different colours) with different biases and precisions.

5.4 Regression model

In this section, we develop a variant of the model proposed in Section 5.3 for regression problems. We shall start by describing the proposed model with a special focus on the how to handle multiple annotators with different biases and reliabilities when the target variables are continuous. Next, we present a variational inference algorithm, highlighting the differences to the classification version. Finally, we show how to optimize the model parameters.

5.4.1 Proposed model

Despite the considerable amount of approaches for learning classifiers from the noisy answers of multiple annotators (see Section 3.1), for continuous response variables this problem has been approached in a much smaller extent. For example, Groot et al. (2011) address this problem in the context of Gaussian processes. In their work, the authors assign a different variance to the likelihood of the data points provided by the different annotators, thereby allowing them to have different noise levels, which can be estimated by maximizing the marginal likelihood of the data. Similarly, the authors in (Raykar et al., 2010) propose an extension of their classification approach to regression problems by assigning different variances to the Gaussian noise models of the different annotators. In this section, we take this idea one step further by also considering a per-annotator bias parameter, which gives the proposed model the ability to overcome certain personal tendencies in the annotators labeling styles that are quite common, for example, in product ratings and document reviews. Furthermore, we empirically validate the proposed model using real multi-annotator data obtained from Amazon mechanical turk (see Section 5.5). This contrasts with the previously mentioned works, which rely only simulated annotators.

For developing a multi-annotator supervised topic model for regression, we shall

follow a similar intuition as the one we considered for classification. Namely, we shall assume that, for a given document d, each annotator provides a noisy version, $y^{d,r} \in \mathbb{R}$, of the true (continuous) target variable, which we denote by $x^d \in \mathbb{R}$. This can be, for example, the true rating of a product or the true sentiment of a document. Assuming that each annotator r has its own personal bias b^r and precision p^r (inverse variance), and assuming a Gaussian noise model for the annotators' answers, we have that

$$y^{d,r} \sim \mathcal{N}(y^{d,r}|x^d + b^r, 1/p^r). \tag{5.16}$$

This approach is therefore more powerful than previous works (Raykar et al., 2010; Groot et al., 2011), where a single precision parameter was used to model the annotators' expertise. Figure 5.7 illustrates this intuition for 4 annotators, represented by different colours. The "green annotator" is the best one, since he is right on the target and his answers vary very little (low bias, high precision). The "yellow annotator" has a low bias, but his answers are very uncertain, as they can vary a lot. Contrarily, the "blue annotator" is very precise, but consistently over-estimates the true target (high bias, high precision). Finally, the "red annotator" corresponds to the worst kind of annotator (high bias and low precision).

Having specified a model for annotators answers given the true targets, the only thing left is to do is to specify a model of the latent true targets x^d given the empirical topic mixture distributions $\bar{\mathbf{z}}^d$. For this, we shall keep things simple and assume a linear model as in sLDA (Mcauliffe and Blei, 2008). The generative process of the proposed model for continuous target variables can then be summarized as follows:

1. For each annotator r
   (a) For each class c
      i. Draw annotator reliability parameter $\boldsymbol{\pi}_c^r|\omega \sim \text{Dirichlet}(\boldsymbol{\pi}_c^r|\omega\mathbf{1}_C)$
2. For each topic k
   (a) Draw topic's distribution over words $\boldsymbol{\beta}_k|\tau \sim \text{Dirichlet}(\boldsymbol{\beta}_k|\tau\mathbf{1}_V)$
3. For each document d
   (a) Draw topic proportions $\boldsymbol{\theta}^d|\alpha \sim \text{Dirichlet}(\boldsymbol{\theta}^d|\alpha\mathbf{1}_K)$
   (b) For the n^{th} word
      i. Draw topic assignment $z_n^d|\boldsymbol{\theta}^d \sim \text{Multinomial}(z_n^d|\boldsymbol{\theta}^d)$
      ii. Draw word $w_n^d|z_n^d, \boldsymbol{\beta}_{1:K} \sim \text{Multinomial}(w_n^d|\boldsymbol{\beta}_{z_n^d})$
   (c) Draw latent (true) value $x^d|\mathbf{z}^d, \boldsymbol{\eta}, \sigma \sim \mathcal{N}(x^d|\boldsymbol{\eta}^{\mathrm{T}}\bar{\mathbf{z}}^d, \sigma^2)$
   (d) For each annotator r
      i. Draw annotator's answer $y^{d,r}|x^d, b^r, p^r \sim \mathcal{N}(y^{d,r}|x^d + b^r, 1/p^r)$

Figure 5.8 shows a graphical representation of the proposed model.

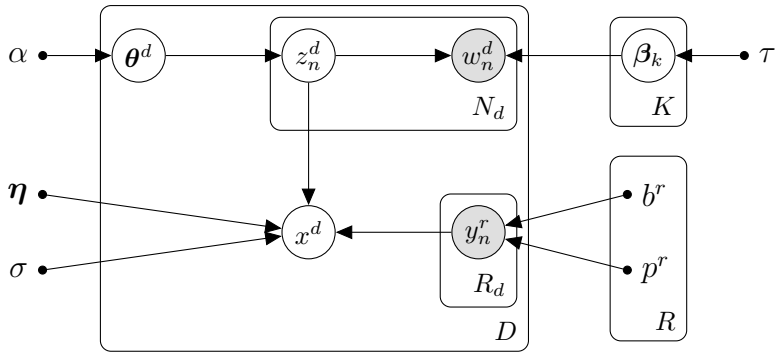


Figure 5.8: Graphical representation of the proposed model for regression.

5.4.2 Approximate inference

The goal of inference is to compute the posterior distribution of the per-document topic proportions $\boldsymbol{\theta}^d$, the per-word topic assignments z_n^d, the per-topic distribution over words $\boldsymbol{\beta}_k$ and the per-document latent true targets x^d. As we did for the classification model, we shall develop a variational inference algorithm using coordinate ascent.

According to the graphical model, the joint distribution of the proposed regression model factorizes as

$$p(\boldsymbol{\theta}_{1:D},\mathbf{z}_{1:D},\mathbf{x},\boldsymbol{\beta}_{1:K},\boldsymbol{\Pi}_{1:R},\mathbf{W},\mathbf{Y}|\Theta) = \left(\prod_{i=1}^{K} p(\boldsymbol{\beta}_i|\tau)\right)\prod_{d=1}^{D} p(\boldsymbol{\theta}^d|\alpha) \times \left(\prod_{n=1}^{N_d} p(z_n^d|\boldsymbol{\theta}^d)\, p(w_n^d|z_n^d,\boldsymbol{\beta}_{1:K})\right) p(x^d|\mathbf{z}^d,\boldsymbol{\eta}) \prod_{r=1}^{R} p(y^{d,r}|x^d,b^r,p^r), \quad (5.17)$$

where $\mathbf{x} = \{x^d\}_{d=1}^D$ and $\Theta = \{\alpha, \tau, \omega, \boldsymbol{\eta}, \mathbf{b}, \mathbf{p}\}$ denotes the model parameters. Notice that the model parameters now include the biases $\mathbf{b} = \{b^r\}_{r=1}^R$ and precisions $\mathbf{p} = \{p^r\}_{r=1}^R$ of the different annotators.

We assume a fully-factorized (mean-field) variational distribution q of the form

$$q(\boldsymbol{\theta}, \mathbf{z}_{1:D}, \boldsymbol{c}, \boldsymbol{\beta}) = \left(\prod_{i=1}^{K} q(\boldsymbol{\beta}_i|\boldsymbol{\zeta}_i)\right)\prod_{d=1}^{D} q(\boldsymbol{\theta}^d|\boldsymbol{\gamma}^d)\left(\prod_{n=1}^{N_d} q(z_n^d|\boldsymbol{\phi}_n^d)\right) q(x^d|m^d, v^d), \quad (5.18)$$

where $\boldsymbol{\zeta}_{1:K}$, $\boldsymbol{\gamma}_{1:D}$, $\boldsymbol{\phi}_{1:D}$, $\mathbf{m} = \{m^d\}_{d=1}^D$ and $\mathbf{v} = \{v^d\}_{d=1}^D$ are the variational parameters. Kindly notice the new Gaussian term, $q(x^d|m^d, v^d)$, corresponding to the approximate posterior distribution of the unobserved true targets.

The lower-bound on the log marginal likelihood is now given by

$$\mathcal{L}(\boldsymbol{\gamma},\boldsymbol{\phi}_{1:D},\mathbf{m},\mathbf{v},\boldsymbol{\zeta}|\Theta) = \mathbb{E}_q[\log p(\boldsymbol{\theta},\mathbf{z}_{1:D},\mathbf{x},\boldsymbol{\beta},\mathbf{W},\mathbf{Y}|\Theta)] - \mathbb{E}_q[\log q(\boldsymbol{\theta},\mathbf{z}_{1:D},\mathbf{x},\boldsymbol{\beta})].$$

Replacing $p(\boldsymbol{\theta},\mathbf{z}_{1:D},\mathbf{x},\boldsymbol{\beta},\mathbf{W},\mathbf{Y}|\Theta)$ and $q(\boldsymbol{\theta},\mathbf{z}_{1:D},\mathbf{x},\boldsymbol{\beta})$ by their definitions in (5.17)

and (5.18) gives

$$\begin{aligned}\mathcal{L}(\boldsymbol{\gamma}, \boldsymbol{\phi}_{1:D}, \mathbf{m}, \mathbf{v}, \boldsymbol{\zeta}|\Theta) = &\sum_{i=1}^{K} \mathbb{E}_q[\log p(\boldsymbol{\beta}_i|\tau)] + \sum_{d=1}^{D} \Bigg(\mathbb{E}_q[\log p(\boldsymbol{\theta}^d|\alpha)] \\ &+ \sum_{n=1}^{N^d} \mathbb{E}_q[\log p(z_n^d|\boldsymbol{\theta}^d)] + \sum_{n=1}^{N^d} \mathbb{E}_q[\log p(w_n^d|z_n^d, \boldsymbol{\beta}_{1:K})] \\ &+ \mathbb{E}_q[\log p(x^d|\bar{\mathbf{z}}^d, \boldsymbol{\eta})] + \sum_{r=1}^{R} \mathbb{E}_q[\log p(y^{d,r}|x^d, b^r, p^r] \Bigg) \\ &- \sum_{i=1}^{K} \mathbb{E}_q[\log q(\boldsymbol{\beta}_i|\boldsymbol{\zeta}_i)] - \sum_{d=1}^{D} \Bigg(\mathbb{E}_q[\log q(\boldsymbol{\theta}^d|\boldsymbol{\gamma}^d)] \\ &- \sum_{n=1}^{N^d} \mathbb{E}_q[\log q(z_n^d|\boldsymbol{\phi}_n^d)] - \mathbb{E}_q[\log q(x^d|m^d, v^d)] \Bigg). \end{aligned} \tag{5.19}$$

Optimizing the evidence lower bound $\mathcal{L}$ w.r.t. γ and ζ yields the same updates from Eqs. 5.7 and 5.8. Optimizing w.r.t. ϕ gives a similar update to the one in sLDA (Mcauliffe and Blei, 2008)

$$\begin{aligned}\phi_{n,i}^d \propto \exp \Bigg(&\Psi(\gamma_i) + \sum_{j=1}^{V} w_{n,j}^d \Bigg(\Psi(\zeta_{i,j}) - \Psi\Bigg(\sum_{k=1}^{V} \zeta_{i,k} \Bigg) \Bigg) \\ &+ \frac{m^d}{N^d \sigma^2} \eta_i - \frac{2(\boldsymbol{\eta}^{\mathrm{T}} \boldsymbol{\phi}_{-n}^d)\eta_i + \eta_i^2}{2(N^d)^2 \sigma^2} \Bigg), \end{aligned} \tag{5.20}$$

where we defined $\boldsymbol{\phi}_{-n}^d \triangleq \sum_{m \neq n} \boldsymbol{\phi}_m^d$. This update differs only from the one in (Mcauliffe and Blei, 2008) by replacing the true target variable by its expected value under the variational distribution, which is given by $\mathbb{E}_q[x^d] = m^d$.

The only variables left for doing inference on are then the unobserved true targets $\mathbf{x} = \{x^d\}_{d=1}^D$. The variational distribution of x^d is governed by two parameters: a mean m^d and a variance v^d. Collecting all the terms in $\mathcal{L}$ that contain m gives

$$\begin{aligned}\mathcal{L}_{[m]} = &- \sum_{d=1}^{D} \sum_{r=1}^{R_d} \frac{p^r}{2} \Big((m^d)^2 + 2m^d b^r - 2y^{d,r} m^d \Big) \\ &- \sum_{d=1}^{D} \frac{1}{2\sigma^2} \Big((m^d)^2 - 2m^d (\boldsymbol{\eta}^{\mathrm{T}} \bar{\boldsymbol{\phi}}^d) \Big). \end{aligned} \tag{5.21}$$

Taking derivatives of $\mathcal{L}_{[m]}$ and setting them to zero gives the following update for the mean m^d of latent true target of the d^{th} document

$$m^d = \frac{\sigma^{-2}(\boldsymbol{\eta}^{\mathrm{T}} \bar{\boldsymbol{\phi}}^d) + \sum_{r=1}^{R_d} p^r (y^{d,r} - b^r)}{\sigma^{-2} + \sum_{r=1}^{R} p^r}. \tag{5.22}$$

Notice how the value of m^d is a weighted average of what the linear regression model on the empirical topic mixture believes that the true target should be, and the bias-corrected answers of the different annotators weighted by their individual precisions.

As for m, we can optimize $\mathcal{L}$ w.r.t. v by collecting all terms that contain v

$$\mathcal{L}_{[v]} = \sum_{d=1}^{D} \left(\frac{1}{2} \log(v^d) - \sum_{r=1}^{R_d} \frac{p^r v^d}{2} - \frac{v^d}{2\sigma^2} \right), \tag{5.23}$$

and taking derivatives, yielding the update

$$v^d = \sigma^2 + \sum_{r=1}^{R_d} \frac{1}{p^r}. \tag{5.24}$$

5.4.3 Parameter estimation

The parameters of the proposed regression model are $\Theta = \{\alpha, \tau, \boldsymbol{\eta}, \sigma, \mathbf{b}, \mathbf{p}\}$. As we did for the classification model, we shall assume the Dirichlet parameters, α and τ, to be fixed. Similarly, we shall assume that the variance of the true targets, σ^2, to be constant. The only parameters left to estimate are then the regression coefficients $\boldsymbol{\eta}$ and the annotators biases, $\mathbf{b} = \{b^r\}_{r=1}^{R}$, and precisions, $\mathbf{p} = \{p^r\}_{r=1}^{R}$, which we estimate using variational Bayesian EM (VBEM).

Since the latent true targets are now linear functions of the documents' empirical topic mixtures (i.e. there is no softmax function), we can find a closed-form solution for the regression coefficients $\boldsymbol{\eta}$. Taking derivatives of $\mathcal{L}$ w.r.t. $\boldsymbol{\eta}$ and setting them to zero, gives the following solution for $\boldsymbol{\eta}$

$$\boldsymbol{\eta}^{\mathrm{T}} = \sum_{d=1}^{D} \mathbb{E}_q\left[\bar{\mathbf{z}}^d (\bar{\mathbf{z}}^d)^{\mathrm{T}}\right]^{-1} (\bar{\boldsymbol{\phi}}^d)^{\mathrm{T}} m^d, \tag{5.25}$$

where

$$\mathbb{E}_q\left[\bar{\mathbf{z}}^d (\bar{\mathbf{z}}^d)^{\mathrm{T}}\right] = \frac{1}{(N^d)^2} \left(\sum_{n=1}^{N^d} \sum_{m \neq n}^{N^d} \boldsymbol{\phi}_n^d (\boldsymbol{\phi}_m^d)^{\mathrm{T}} + \sum_{n=1}^{N^d} \mathrm{diag}(\boldsymbol{\phi}_n^d) \right).$$

We can find maximum likelihood estimates for the annotator biases b^r by optimizing the lower bound on the marginal likelihood. The terms in $\mathcal{L}$ that involve b are

$$\mathcal{L}_{[b]} = \sum_{d=1}^{D} \sum_{r=1}^{R_d} \frac{p^r}{2} \left(2y^{d,r} b^r - 2m^d b^r - (b^r)^2 \right). \tag{5.26}$$

Taking derivatives with respect to b^r gives the following estimate for the bias of the r^{th} annotator

$$b^r = \frac{1}{D_r} \sum_{d=1}^{D_r} \left(y^{d,r} - m^d \right). \tag{5.27}$$

Similarly, we can find maximum likelihood estimates for the precisions p^r of the different annotators by considering only the terms in $\mathcal{L}$ that contain p

$$\mathcal{L}_{[p]} = \sum_{d=1}^{D} \sum_{r=1}^{R_d} \left(\frac{1}{2} \log(p^r) - \frac{p^r v^d}{2} - \frac{p^r}{2} (y^{d,r} - m^d - b^r)^2 \right). \tag{5.28}$$

Algorithm 2 Stochastic variational inference for the proposed regression model

Initialize $\boldsymbol{\gamma}_{1:D}^{(0)}$, $\boldsymbol{\phi}_{1:D}^{(0)}$, $\mathbf{m}^{(0)}$, $\mathbf{v}^{(0)}$, $\boldsymbol{\zeta}_{1:K}^{(0)}$, $t = 0$
repeat
  Set t = t + 1
  Sample a document $\mathbf{w}^d$ uniformly from the corpus
  repeat
    Compute $\boldsymbol{\phi}_n^d$ using (5.20), for $n \in \{1..N_d\}$
    Compute $\boldsymbol{\gamma}^d$ using (5.7)
    Compute m^d using (5.22)
    Compute v^d using (5.24)
  until local parameters $\boldsymbol{\phi}_n^d$, $\boldsymbol{\gamma}^d$, m^d and v^d converge
  Compute step-size $\rho_t = (t + delay)^{-\kappa}$
  Update topics variational parameters

$$\zeta_{i,j}{}^{(t)} = (1 - \rho_t)\zeta_{i,j}^{(t-1)} + \rho_t\left(\tau + D\sum_{n=1}^{N_d} w_{n,j}^d \phi_{n,i}^d\right)$$

until global convergence criterion is met

The maximum likelihood estimate for the precision (inverse variance) of the r^{th} annotator is then given by

$$p^r = \left(\frac{1}{D_r}\sum_{d=1}^{D_r}\left(v^d + (y^{d,r} - m^d - b^r)^2\right)\right)^{-1}. \tag{5.29}$$

Given a set of fitted parameters, it is then straightforward to make predictions for new documents: it is just necessary to infer the (approximate) posterior distribution over the word-topic assignments z_n^d for all the words using the coordinate ascent updates of standard LDA (Eqs. 5.7 and 5.14), and then use the mean topic assignments $\bar{\boldsymbol{\phi}}^d$ to make predictions $x_*^d = \boldsymbol{\eta}^{\mathrm{T}}\bar{\boldsymbol{\phi}}^d$.

5.4.4 Stochastic variational inference

As we did for the classification model from Section 5.3, we can envision developing a stochastic variational inference for the proposed regression model. In this case, the only "global" latent variables are the per-topic distributions over words, $\boldsymbol{\beta}_k$. As for the "local" latent variables, ignoring the words' topic assignments ϕ_n^d, instead of a single variable $\boldsymbol{\lambda}^d$, we now have two variables per-document: m^d and v^d. The stochastic variational inference can then be summarized as shown in Algorithm 2. For added efficiency, one can also perform stochastic updates of the annotators biases b^r and precisions p^r, by taking a step in the direction of the gradient of the noisy evidence lower bound scaled by the step-size ρ_t.

5.5 Experiments

In this section, the proposed multi-annotator supervised LDA models for classification and regression (MA-sLDAc and MA-sLDAr, respectively) are validated using

both simulated annotators on popular corpora and using real multiple-annotator labels obtained from Amazon mechanical turk.[2] Namely, we shall consider the following real-world problems:

- classifying posts and news stories;
- classifying images according to their content;
- predicting the number of stars that a given user gave to a restaurant based on the review;
- predicting movie ratings using the text of the reviews.

We will start by evaluating the classification model proposed in Section 5.3 in the first two problems (see Section 5.5.1) and use the last two regression problems for evaluating the model proposed in Section 5.4 (see Section 5.5.2).

5.5.1 Classification

Simulated annotators

In order to first validate the proposed model for classification problems in a slightly more controlled environment, the well-known 20-Newsgroups benchmark corpus (Lang, 1995) was used by simulating multiple annotators with different levels of expertise. The 20-Newsgroups consists of twenty thousand messages taken from twenty newsgroups, and is divided in six super-classes, which are, in turn, partitioned in several sub-classes. For this first set of experiments, only the four most populated super-classes were used, namely "computers", "science", "politics" and "recreative". The preprocessing of the documents consisted of stemming and stopwords removal. After that, 75% of the documents were randomly selected for training and the remaining 25% for testing.

The different annotators were simulated by sampling their answers from a multinomial distribution, where the parameters are given by the lines of the annotators' confusion matrices. Hence, for each annotator r, we start by pre-defining a confusion matrix $\boldsymbol{\pi}^r$ with elements $\pi^r_{c,l}$, which correspond to the probability that the annotators' answer is l given that the true label is c, i.e. $p(y^r = l|c)$. Then, the answers are sampled i.i.d. from $y^r \sim \text{Multinomial}(y^r|\pi^r_{c,l})$. This procedure was used to simulate 5 different annotators with the following accuracies: 0.737, 0.468, 0.284, 0.278, 0.260. In this experiment, no repeated labelling was used. Hence, each annotator only labels roughly one-fifth of the data. When compared to the ground truth, the simulated answers revealed an accuracy of 0.405. See Table 5.3 for an overview of the details of the classification datasets used.

Both the *batch* and the stochastic variational inference (*svi*) versions of the proposed model (MA-sLDAc) are compared with the following baselines:

- *LDA + LogReg (mv)*: This baseline corresponds to applying unsupervised LDA to the data, and learning a logistic regression classifier on the inferred topic distributions of the documents. The labels from the different annotators were aggregated using majority voting (mv). Notice that, when there is a single

[2]Source code and datasets used are available at: http://amilab.dei.uc.pt/fmpr/ma-slda/

Dataset	Num. classes	Train/test sizes	Num. answers per instance (± stddev.)	Mean annotators accuracy (± stddev.)	Maj. vot. accuracy
20 Newsgroups	4	11536/3846	1.000 ± 0.000	0.405 ± 0.182	0.405
Reuters-21578	8	1800/5216	3.007 ± 1.019	0.568 ± 0.262	0.710
LabelMe	8	1000/1688	2.547 ± 0.576	0.692 ± 0.181	0.769

Table 5.3: Overall statistics of the classification datasets used in the experiments.

annotator label per instance, majority voting is equivalent to using that label for training. This is the case of the 20-Newsgroups' simulated annotators, but the same does not apply for the experiments with AMT.

- *LDA + Raykar*: For this baseline, the model of Raykar et al. (2010) was applied using the documents' topic distributions inferred by LDA as features.
- *LDA + Rodrigues*: This baseline is similar to the previous one, but uses the model of Rodrigues et al. (2013a) instead.
- *Blei 2003 (mv)*: The idea of this baseline is to replicate a popular state-of-the-art approach for document classification. Hence, the approach of Blei et al. (2003) was used. It consists of applying LDA to extract the documents' topic distributions, which are then used to train a support vector machine (SVM). Similarly to the previous approach, the labels from the different annotators were aggregated using majority voting (mv).
- *sLDA (mv)*: This corresponds to using the classification version of sLDA (Wang et al., 2009) with the labels obtained by performing majority voting (mv) on the annotators' answers.

For all the experiments the hyper-parameters α, τ and ω were set using a simple grid search in the collection $\{0.01, 0.1, 1.0, 10.0\}$. The same approach was used to optimize the hyper-parameters of the all the baselines. For the *svi* algorithm, different mini-batch sizes and forgetting rates κ were tested. For the 20-Newsgroup dataset, the best results were obtained with a mini-batch size of 500 and $\kappa = 0.6$. The *delay* was kept at 1. The results are shown in Figure 5.9 for different numbers of topics, where we can see that the proposed model outperforms all the baselines, being the *svi* version the one that performs best.

In order to assess the computational advantages of the stochastic variational inference (*svi*) over the *batch* algorithm, the log marginal likelihood (or log evidence) was plotted against the number of iterations. Figure 5.10 shows this comparison. Not surprisingly, the *svi* version converges much faster to higher values of the log marginal likelihood when compared to the *batch* version, which reflects the efficiency of the *svi* algorithm.

Amazon mechanical turk

In order to validate the proposed classification model in real crowdsourcing settings, Amazon mechanical turk (AMT) was used to obtain labels from multiple annotators for two popular datasets: Reuters-21578 (Lewis, 1997) and LabelMe (Russell et al., 2008).

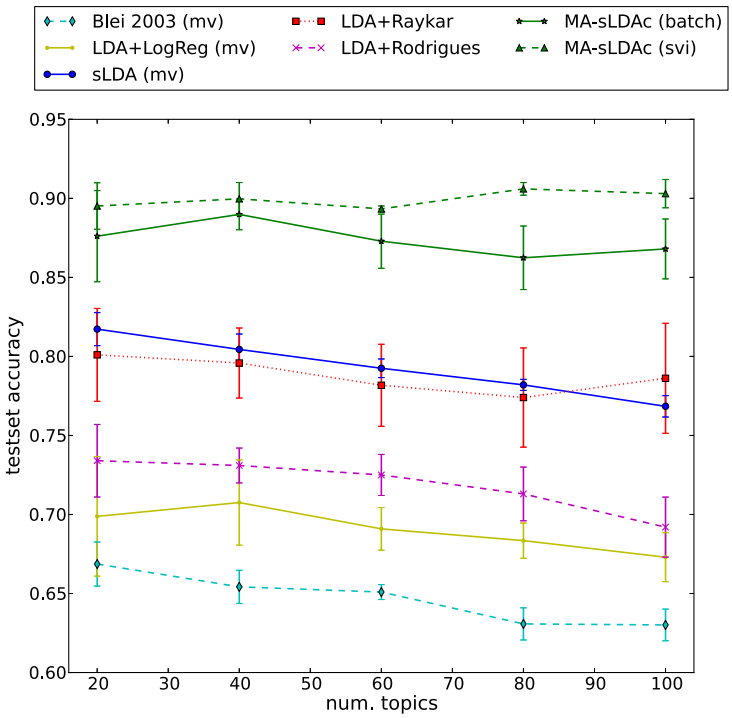


Figure 5.9: Average testset accuracy (over 5 runs; ± stddev.) of the different approaches on the 20-Newsgroups data.

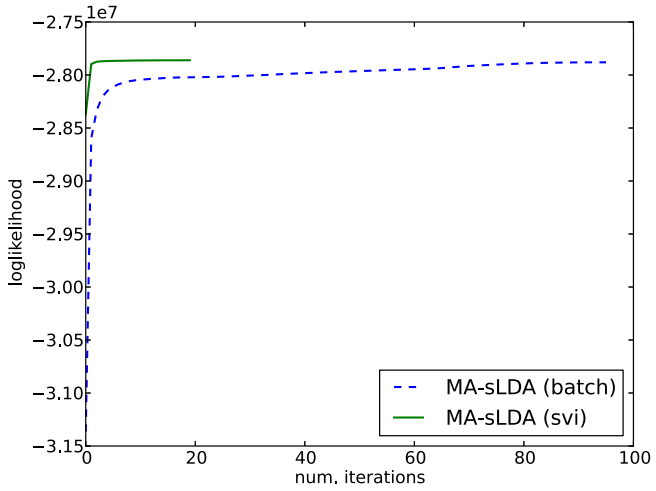


Figure 5.10: Comparison of the log marginal likelihood between the *batch* and the stochastic variational inference (*svi*) algorithms on the 20-Newsgroups corpus.

The Reuters-21578 is a collection of manually categorized newswire stories with labels such as Acquisitions, Crude-oil, Earnings or Grain. For this experiment, only the documents belonging to the ModApte split were considered with the additional constraint that the documents should have no more than one label. This resulted in a total of 7016 documents distributed among 8 classes. Of these, 1800 documents were submitted to AMT for multiple annotators to label, giving an average of approximately 3 answers per document (see Table 5.3 for further details). The

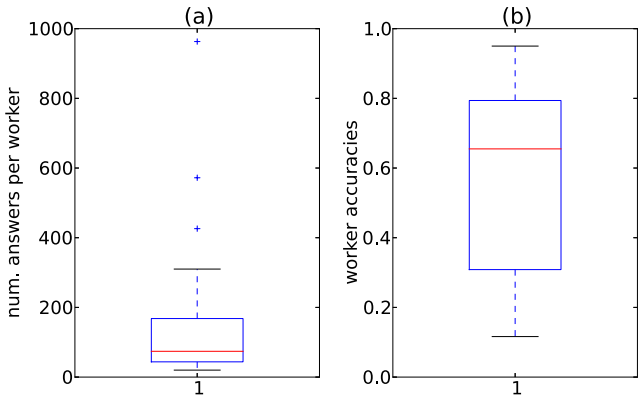


Figure 5.11: Boxplot of the number of answers per worker (a) and their respective accuracies (b) for the Reuters dataset.

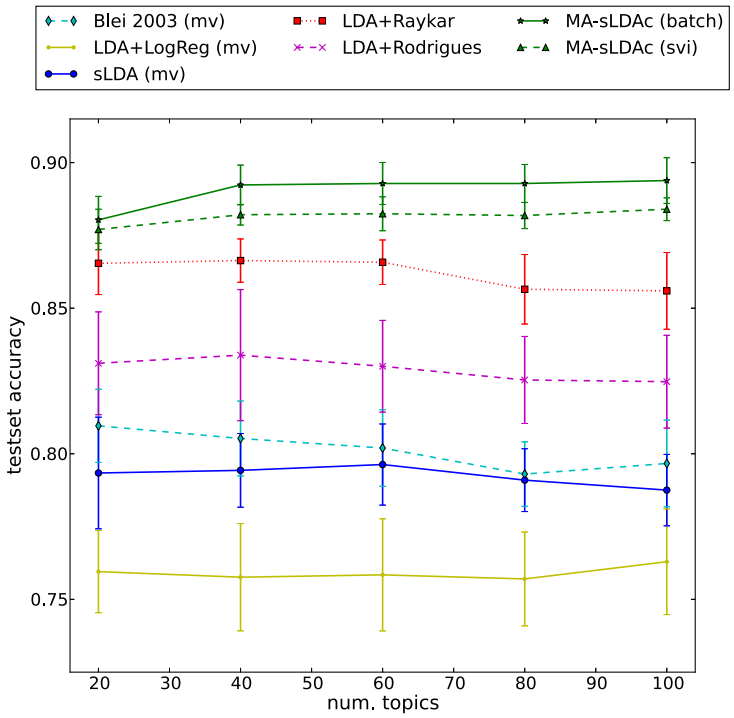


Figure 5.12: Average testset accuracy (over 30 runs; ± stddev.) of the different approaches on the Reuters data.

remaining 5216 documents were used for testing. The collected answers yield an average worker accuracy of 56.8%. Applying majority voting to these answers reveals a ground truth accuracy of 71.0%. Figure 5.11 shows the boxplots of the number of answers per worker and their accuracies. Observe how applying majority voting yields a higher accuracy than the median accuracy of the workers.

The results obtained by the different approaches are given in Figure 5.12, where it can be seen that the proposed model (MA-sLDAc) outperforms all the other approaches. For this dataset, the *svi* algorithm is using mini-batches of 300 documents.

The proposed model was also validated using a dataset from the computer vision

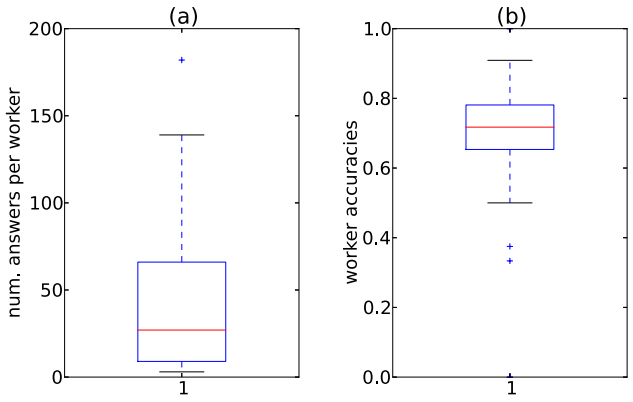


Figure 5.13: Boxplot of the number of answers per worker (a) and their respective accuracies (b) for the LabelMe dataset.

domain: LabelMe (Russell et al., 2008). In contrast to the Reuters and Newsgroups corpora, LabelMe is an open online tool to annotate images. Hence, this experiment allows us to see how the proposed model generalises beyond non-textual data. Using the Matlab interface provided in the projects' website, we extracted a subset of the LabelMe data, consisting of all the 256 x 256 images with the categories: "highway", "inside city", "tall building", "street", "forest", "coast", "mountain" or "open country". This allowed us to collect a total of 2688 labeled images. Of these, 1000 images were given to AMT workers to classify with one of the classes above. Each image was labeled by an average of 2.547 workers, with a mean accuracy of 69.2%. When majority voting is applied to the collected answers, a ground truth accuracy of 76.9% is obtained. Figure 5.13 shows the boxplots of the number of answers per worker and their accuracies. Interestingly, the worker accuracies are much higher and their distribution is much more concentrated than on the Reuters-21578 data (see Figure 5.11), which suggests that this is an easier task for the AMT workers.

The preprocessing of the images used is similar to the approach of Fei-Fei and Perona (2005). It uses 128-dimensional SIFT (Lowe, 1999) region descriptors selected by a sliding grid spaced at one pixel. This sliding grid extracts local regions of the image with sizes uniformly sampled between 16 x 16 and 32 x 32 pixels. The 128-dimensional SIFT descriptors produced by the sliding window are then fed to a k-means algorithm (with $k = 200$) in order construct a vocabulary of 200 "visual words". This allows us to represent the images with a bag of visual words model.

With the purpose of comparing the proposed model with a popular state-of-the-art approach for image classification, for the LabelMe dataset, the following baseline was introduced:

- *Bosch 2006 (mv)*: This baseline is similar to one in Bosch et al. (2006). The authors propose the use of pLSA to extract the latent topics, and the use of k-nearest neighbor (kNN) classifier using the documents' topics distributions. For this baseline, unsupervised LDA is used instead of pLSA, and the labels from the different annotators for kNN (with $k = 10$) are aggregated using majority voting (mv).

The results obtained by the different approaches for the LabelMe data are shown in Figure 5.14, where the *svi* version is using mini-batches of 200 documents.

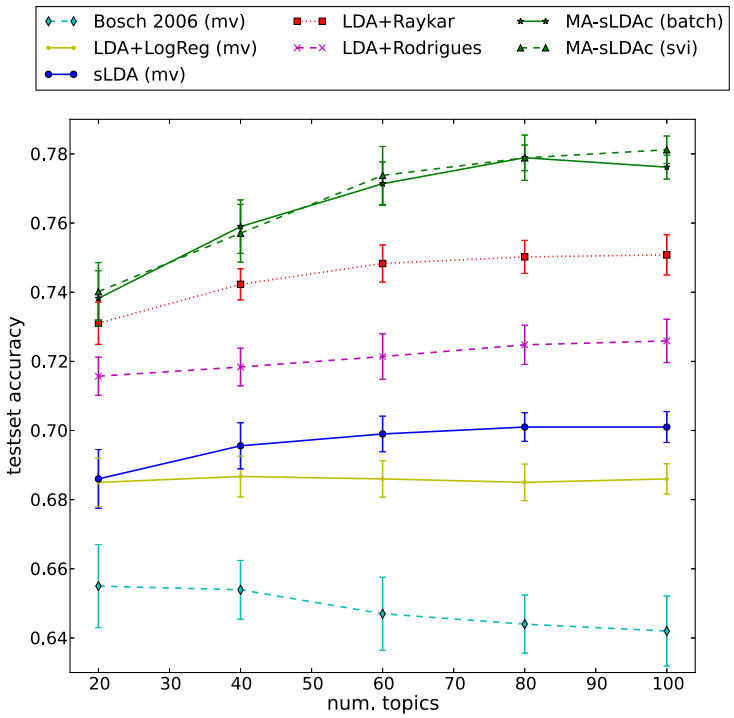
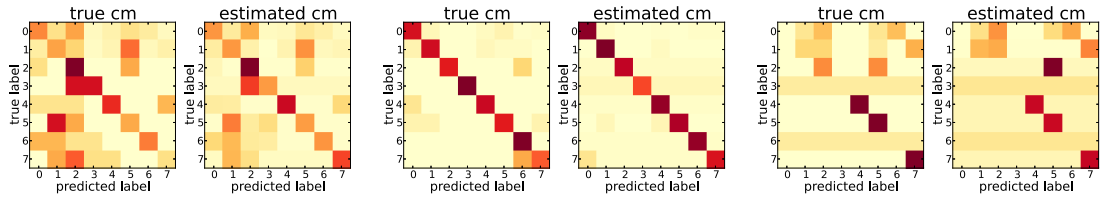


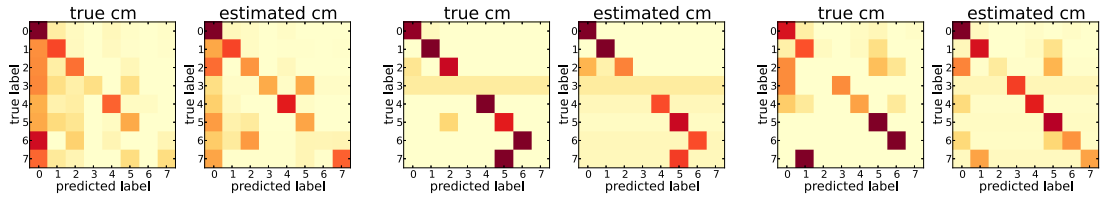
Figure 5.14: Average testset accuracy (over 30 runs; ± stddev.) of the different approaches on the LabelMe data.

Analyzing the results for the Reuters-21578 and LabelMe data, we can observe that MA-sLDAc outperforms all the baselines, with slightly better accuracies for the *batch* version, especially in the Reuters data. Interestingly, the second best results are consistently obtained by the multi-annotator approaches, which highlights the need for accounting for the noise and biases of the answers of the different annotators. All of these are conclusions readable from the figures. Nevertheless, we are interested in assessing the statistical significance of the results obtained. In order to do so, we selected the accuracies of the different models for the number of topics that produced the best results for the most competitive baseline (LDA+Raykar). Using this data, we first used a Kolmogorov-Smirnov test to verify that there were statistic facts supporting that the data was drawn from a normal distribution. Then, a paired t-test was used to compare that batch version of MA-sLDAc with LDA+Raykar for the three datasets considered. The highest p-value obtained was 8×10^{-11}, from which we can conclude that all the differences are significantly different.

In order to verify that the proposed model was estimating the (normalized) confusion matrices $\boldsymbol{\pi}^r$ of the different workers correctly, a random sample of them was plotted against the true confusion matrices (i.e. the normalized confusion matrices evaluated against the true labels). Figs. 5.15 and 5.16 show the results obtained with 60 topics, where the colour intensity of the cells increases with the magnitude of the value of $p(y^{d,r} = l|c^d) = \pi^r_{c,l}$. Using this visualization we can verify that the AMT workers are quite heterogeneous in their labeling styles and in the kind of mistakes they make, with several workers showing clear biases (e.g. workers 3 and 4 in Figure 5.15, and workers 1 and 5 in Figure 5.16), while others made mistakes more randomly (e.g. worker 1 in Figure 5.15, and worker 6 in Figure 5.16). Nevertheless, the proposed is able to capture these patterns correctly and account for their effect.

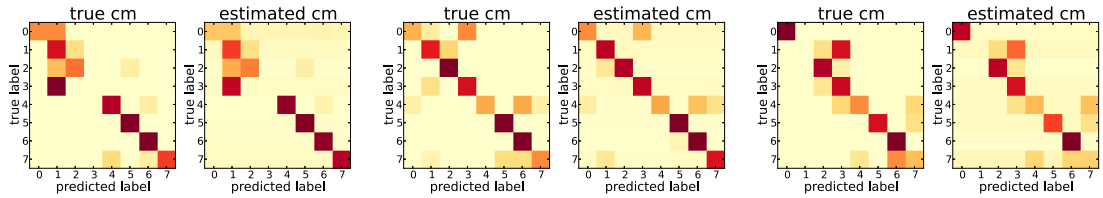


(a) annotator 1 (b) annotator 2 (c) annotator 3

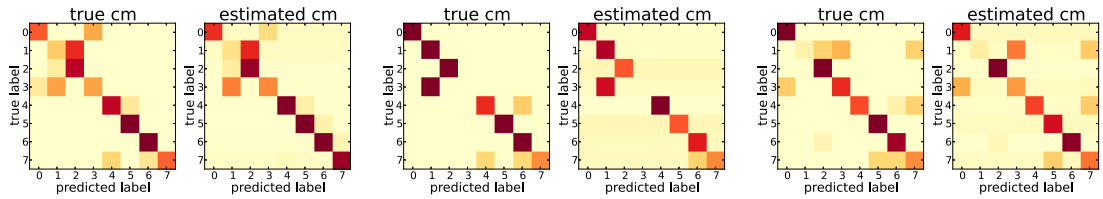


(d) annotator 4 (e) annotator 5 (f) annotator 6

Figure 5.15: True vs. estimated confusion matrix (cm) of 6 different workers of the Reuters-21578 dataset.



(a) annotator 1 (b) annotator 2 (c) annotator 3



(d) annotator 4 (e) annotator 5 (f) annotator 6

Figure 5.16: True vs. estimated confusion matrix (cm) of 6 different workers of the LabelMe dataset.

5.5.2 Regression

Simulated annotators

As for the proposed classification model, we start by validating MA-sLDAr using simulated annotators on a popular corpus where the documents have associated targets that we wish to predict. For this purpose, we shall consider a dataset of user-submitted restaurant reviews from the website we8there.com. This dataset was originally introduced by Mauá and Cozman (2009) and it consists of 6260 reviews. For each review, there is a five-star rating on four specific aspects of quality (food, service, value, and atmosphere) as well as the overall experience. Our goal is then to predict the overall experience of the user based on her comments in the review. We apply the same preprocessing as in (Taddy, 2013), which consists in

Table 5.4: Overall statistics of the regression datasets used in the experiments.

Dataset	Train/test sizes	Num. answers per instance (± stddev.)	Mean annotators R^2 (± stddev.)	Mean answer R^2
we8there	4624/1542	5.000 ± 0.000	-0.525 ± 1.364	0.798
movie reviews	1500/3506	4.960 ± 0.196	-0.387 ± 1.267	0.830

tokenizing the text into bigrams and discarding those that appear in less than ten reviews. The preprocessing of the documents consisted of stemming and stop-words removal. After that, 75% of the documents were randomly selected for training and the remaining 25% for testing.

As with the classification model, we seek to simulate an heterogeneous set of annotators in terms of reliability and bias. Hence, in order to simulate an annotator r, we proceed as follows: let x^d be the true review of the restaurant; we start by assigning a given bias b^r and precision p^r to the reviewers, depending on what type of annotator we wish to simulate (see Figure 5.7); we then sample a simulated answer as $y^{d,r} \sim \mathcal{N}(x^d + b^r, 1/p^r)$. Using this procedure, we simulated 5 annotators with the following (bias, precision) pairs: (0.1, 10), (-0.3, 3), (-2.5, 10), (0.1, 0.5) and (1, 0.25). The goal is to have 2 good annotators (low bias, high precision), 1 highly biased annotator and 2 low precision annotators where one is unbiased and the other is reasonably biased. The coefficients of determination (R^2) of the simulated annotators are: {0.940, 0.785, -2.469, -0.131, -1.749}. Computing the mean of the answers of the different annotators yields a R^2 of 0.798. Table 5.4 gives an overview on the statistics of datasets used in the regression experiments.

We compare the proposed model (MA-sLDAr) with the two following baselines:

- *LDA + LinReg (mean)*: This baseline corresponds to applying unsupervised LDA to the data, and learning a linear regression model on the inferred topic distributions of the documents. The answers from the different annotators are aggregated by computing the mean.
- *sLDA (mean)*: This corresponds to using the regression version of sLDA (Mcauliffe and Blei, 2008) with the target variables obtained by computing the mean of the annotators' answers.

Figure 5.17 shows the results obtained for different numbers of topics. Due to the stochastic nature of both the annotators simulation procedure and the initialization of the variational Bayesian EM algorithm, we repeated each experiment 30 times and report the average R^2 obtained with the corresponding standard deviation. Since the regression datasets that are considered in this chapter are not large enough to justify the use of a stochastic variational inference (*svi*) algorithm, we only made experiments using the *batch* algorithm developed in Section 5.4.2. The results obtained clearly demonstrate the improved performance of MA-sLDAr over the other methods.

Amazon mechanical turk

The proposed multi-annotator regression model (MA-sLDAr) was also validated with real annotators by using Amazon mechanical turk. For that purpose, the movie

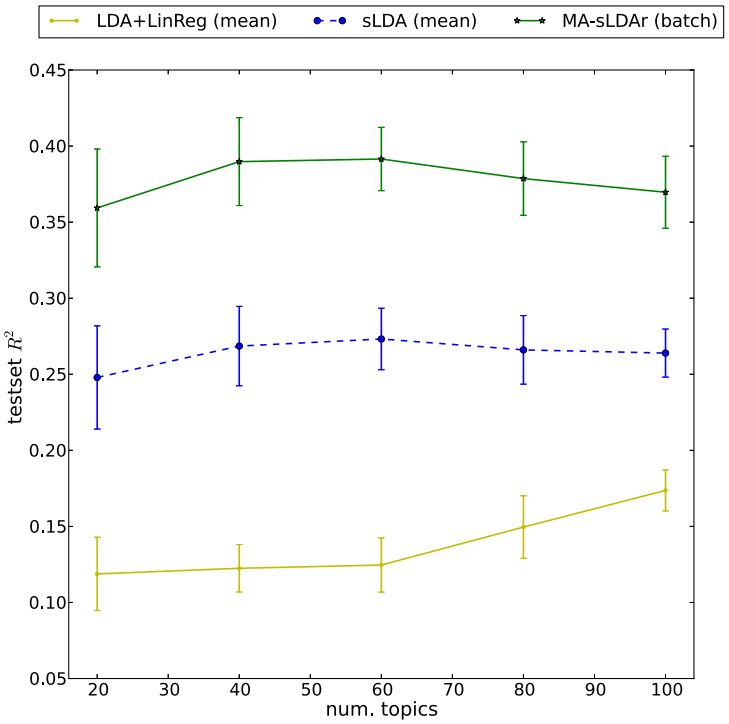


Figure 5.17: Average testset R^2 (over 30 runs; $\pm$ stddev.) of the different approaches on the we8there data.

reviews dataset from (Pang and Lee, 2005) was used. This dataset consists of 5006 movie reviews along with their respective star rating (from 1 to 10). The goal of this experiment is then to predict how much a person liked a movie based on what she says about it. Using AMT, we ask workers to guess how much they think the writer of the review liked the movie based on her comments. An average of 4.96 answers per-review was collected for a total of 1500 reviews. The remaining reviews were used for testing. In average, each worker rated a total of approximately 55 reviews. Using the mean answer as an estimate of the true rating of the movie yields a R^2 of 0.830. Table 5.4 gives an overview of the statistics of this data. Figure 5.18 further shows boxplots of the number of answers per worker, as well as boxplots of their respective biases (b^r) and variances (inverse precisions, $1/p^r$).

The preprocessing of the text consisted of stemming and stop-words removal. Using the preprocessed data, the proposed MA-sLDAr model was compared with the same baselines that were used with the we8there dataset. Figure 5.19 shows the results obtained for different numbers of topics. These results show that the proposed model outperforms all the other baselines. As we did with the classification version, the statistical difference between the proposed model and the best baseline method was analysed using a paired t-test. For simplicity, we focus only on the number of topics for which the best baseline method produces its best results. According to the results of the paired t-test, all the differences were statistically significant on both datasets (we8there and movie reviews), with the highest p-value being 2×10^{-5}.

With the purpose of verifying that the proposed model is indeed estimating the biases and precisions of the different workers correctly, we plotted the true values against the estimates of MA-sLDAr with 60 topics for a random subset of 10 workers. Figure 5.20 shows the obtained results, where higher colour intensities

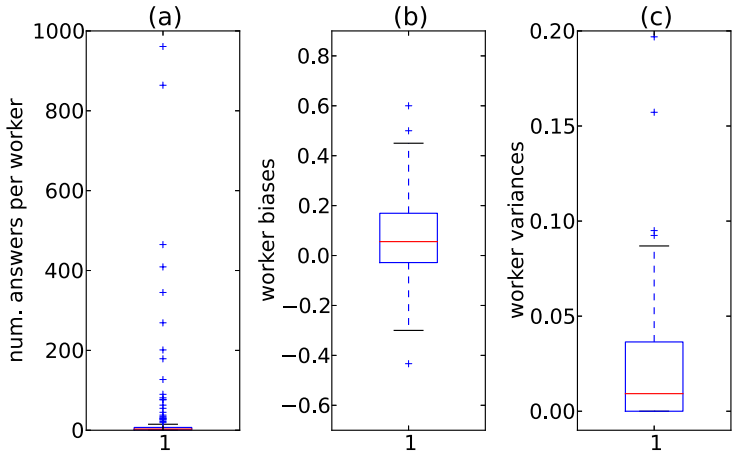


Figure 5.18: Boxplot of the number of answers per worker (a) and their respective biases (b) and variances (c) for the movie reviews dataset.

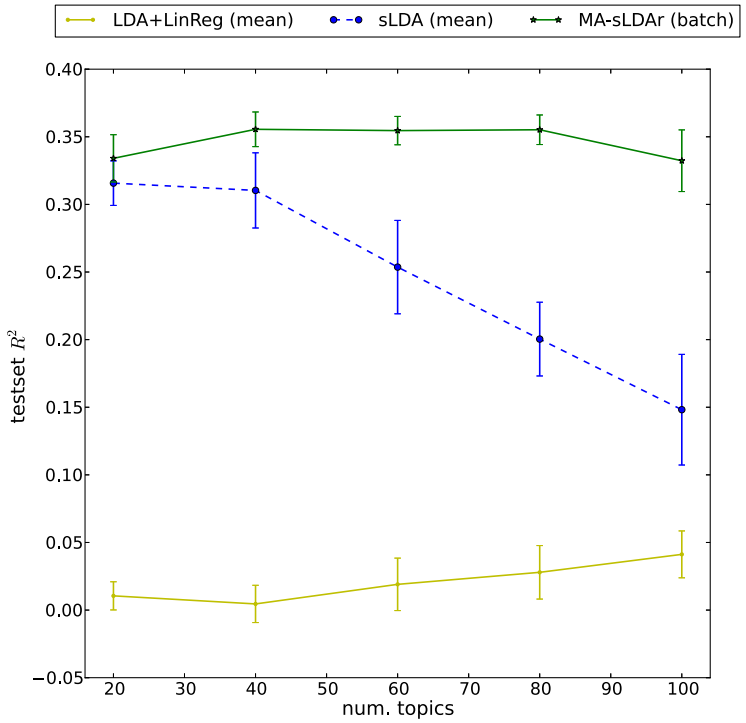


Figure 5.19: Average testset R^2 (over 30 runs; $\pm$ stddev.) of the different approaches on the movie reviews data.

indicate higher values. Ideally, the colour of two horizontally-adjacent squares would then be of similar shades, and this is indeed what happens in practice for the majority of the workers, as Figure 5.20 shows. Interestingly, the figure also shows that there are a couple of workers that are considerably biased (e.g. workers 6 and 8) and that those biases are being correctly estimated, thus justifying the inclusion of a bias parameter in the proposed model, which, as we earlier mentioned, contrasts with previous works (Raykar et al., 2010; Groot et al., 2011).

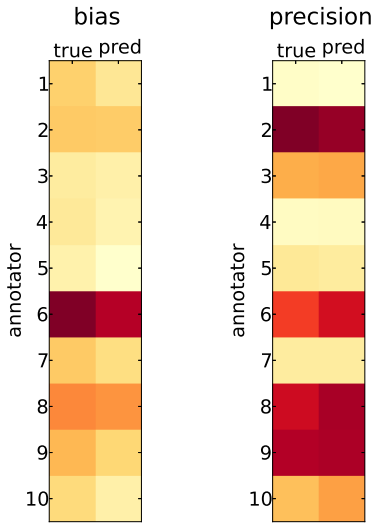


Figure 5.20: True vs. predicted biases and precisions of 10 random workers of the movie reviews dataset.

5.6 Conclusion

In this chapter, we proposed two supervised topic models that are able to learn from multiple annotators and crowds, by accounting for their biases and different levels of expertise. Given the large sizes of modern datasets, and considering that the majority of the tasks for which crowdsourcing and multiple annotators are desirable candidates, generally involve complex high-dimensional data such as text and images, the proposed model constitutes a strong contribution for the multi-annotator paradigm. These models are then capable of jointly modeling the words in documents as arising from a mixture of topics, as well as the latent true target variables and the (noisy) answers of the multiple annotators. We developed two distinct models, one for classification and another for regression, that share similar intuitions but that inevitably differ due the nature of the target variables. We empirically showed, using both simulated and real annotators from Amazon mechanical turk that the proposed models are able to outperform state-of-the-art approaches in several real-world problems, such as classifying posts, news stories and images, or predicting the number of stars of restaurant and the rating of movies based on their reviews. For this, we use various popular datasets from the state of the art, that are commonly used for benchmarking machine learning algorithms. Finally, efficient stochastic variational inference algorithms were described, which give the proposed models the ability to scale to large datasets.

Part II

Using crowds data for understanding urban mobility

Chapter 6

Explaining non-habitual transport overcrowding with internet data

6.1 Introduction

"We are drowning in information and starving for knowledge."
– John Naisbitt

In Part I, we discussed how to learn predictive models from the noisy answers of multiple annotators and crowds. Namely, we saw different ways in which we could improve the model's predictions by accounting for the different levels of expertise of the various annotators, as well as their biases. In this part of the thesis, we shall take a different perspective on the value of crowds' data, by considering how it can be used to improve transportation demand prediction models and to help us understand urban mobility. In particular, we shall consider data produced by crowds regarding special events that take place in the city. This kind of data is vastly available on the web, mostly in textual form. However, turning it into useful knowledge can be very challenging.

During the last decade, pervasive technologies such as radio-frequency identification, global positioning systems (GPS), WiFi, NFC and mobile phone communications have become ubiquitous. These mobility traces are increasingly available for practitioners and researchers to provide a better understanding of urban mobility. For example, using a large cell-phone dataset, Gonzalez et al. (2008) showed that individual mobility travel patterns generally follow a single spatial probability distribution, indicating that, despite their inherent heterogeneity, humans follow simple reproducible patterns. In fact, this asserts the remarkable yet not so surprising fact that human mobility is habitual for the vast majority of the time. This principle has been behind several other works, for example, to estimate disease spreading (Adams and Kapan, 2009) or vehicular network routing protocols (Xue et al., 2009).

Despite other studies that stretch the boundaries of that principle and verify that it is widely persistent (e.g. Song et al. (2010); Jiang et al. (2013)), mobility behaviour heterogeneity is recognized to create predictability challenges. This is particularly important when it involves large crowds. As pointed out by Potier et al. (2003), even for well-known big events (e.g. olympic games), demand is inevitably more difficult to forecast than habitual mobility, particularly in the case of open-gate

events. When facing these constraints, authorities tend to rely on trial and error experience (for recurring events), checklists (e.g. (FHWA)) and sometimes invest in a reactive approach rather than planning, as happens in Germany, with the real-time traffic and traveller information (RTTI) and its active traffic management (Bolte, 2006), and in the Netherlands (Middleham, 2006). However, such tools have limited applicability, particularly for smaller and medium events, which are harder to capture and to evaluate.

Taking advantage of the amount and quality of pervasive technologies such as radio-frequency identification, smartcards, and mobile phone communications, it is then possible to detect crowds in almost real time with very low risk for privacy. By itself, crowd detection can be valuable for safety reasons, as well as for real-time supply/demand management of transportation, communications, food stock, logistics, water, or any other system sensitive to aggregated human behaviour. But, although such technologies help detect and quantify crowds, they have limited power in explaining why they happen. As previous works show (Potier et al., 2003; Jiang et al., 2013), for recurring crowds, such as peak-hour commuting, this explanatory challenge is trivial, but the same cannot be said of non-habitual cases. Without local context knowledge, it is not possible to discern an explanation.

Fortunately, another pervasive technology exists: the internet, which is rich in local context information generated by large online crowds. Information about public special events, such as sports games, concerts, parades, sales, demonstrations and festivals, is abundant, and so are social networks (e.g. Twitter, Facebook) and other platforms that have dynamic contextal content (e.g. news feeds). Using a manually selected subset of events from the Boston Globe website[1] and a massive cell-phone dataset, Calabrese et al. (2010) studied the public home distributions for different types of special events (e.g. sports, concerts, theatre). They identified a strong correlation between public neighborhood distributions and event types. This is a key finding, since it implies that such heterogeneous cases are still predictable as long as we have sufficient event information. They did not, however, consider multiple event interactions or deeper explanatory content (e.g. event description text), as we do in this chapter.

Particularly for public transport operations and management, the treatment of overcrowding depends on understanding why people are there, and where/when they will go next. Only then can the manager react accordingly (e.g. add extra buses, trains, send taxis). For example, by knowing that an overcrowding hotspot is due to a concert, one can also estimate its duration (until about after the concert begins) and a possible next hotspot (after the concert ends). If instead it was due to a series of small scattered events, the treatment may be different (e.g. no single ending hotspot). Maybe even more importantly, by understanding such impacts on a post-hoc analysis, one can also better prepare for the next time that similar events happen.

This chapter proposes to solve the following problem: given a non-habitual large crowd — an *overcrowding hotspot* — what are its potential causes and how do they individually contribute to the overall impact? We will focus particularly on the problem of public transport overcrowding in special events' areas as the main practical motivation and case study. Given the importance of these social phenomena, many traffic management centers have teams of people that are responsible for

[1] http://www.bostonglobe.com

periodically scanning the internet and newspapers in search for special events. In fact, for very large events, this problem is generally solved, albeit manually. The challenge comes when multiple smaller events co-occur in the same area to form a relevant hotspot. It is not only harder to find them but also extremely difficult to intuitively estimate their aggregated impact.

Overcrowding hotspots are identified and measured by analyzing 4 months of public transport data from the city-state of Singapore. During the whole period of the dataset, we collected special events data from five websites, as well as their Facebook likes and Google hits. Hence, while in Part I we considered explicit forms of crowdsourcing, such as Amazon mechanical turk, here we are exploring crowdsourcing in a more implicit form. Our goal, is then to use this crowd-generated data to break each non-habitual overcrowding hotspot into a set of explanatory components, along with estimates of their respective shares in the total overcrowding counts. In order to do so, we propose an additive model, where each hotspot is formalized as a sum of potential explanatory components.

This methodology is applicable beyond the specific case of public transport overcrowding as long as the key research question and ingredients remain. For example, during special events, cell phone, WiFi network, energy, or catering/logistics systems may equally suffer from disruptions. If there is both pervasive and explanatory data to quantify and correlate the impacts, the general procedure remains the same.

6.2 Identifying overcrowding hotspots

There is no golden rule threshold above which we can identify overcrowding. The intuition is that it should happen whenever the supply (e.g. buses) is insufficient to satisfy the demand (e.g. travelers), which leads to very heavily loaded vehicles or, ultimately, to denied boardings. The latter are non-observable from the dataset used, and so are estimates of bus or train loading, therefore we resort to indirect measurements such as the total number of arrivals.

In order to cope with demand fluctuations, transport systems are generally designed with reasonable spare capacity, so we need to define the point above which we consider it under stress. For any given study area and point in time, we define such point to correspond to the 90% percentile, i.e. whenever the number of arrivals exceeds such threshold, we consider that *overcrowding* is occurring. This threshold choice is based on our intuition and experience together with discussions with local experts, not being attached to a strong theory or experimental study. However, the main contribution of this chapter is methodological and all principles should remain the same, either by choosing another threshold or detecting hotspots differently, like, for example, by sensing denied boardings or monitoring bus load.

We quantify the impact by summing up the excess amount of arrivals above the median line in a continuous time frame, discretized by 30-minute intervals. Figure 6.1 visualizes this calculation. On the 24th of December 2012, there were 3 hotspots in this area (Singapore Expo). In fact, there were two simultaneous events during several hours: Megatex, related to IT and electronics; and Kawin-kawin makan-makan 2012, an event about Malay food and lifestyle products.

Whenever hotspots are both short in time and with small relative impact (e.g. below 5% of the mean or just 30 minutes in duration), we remove them as they should not represent a problem from a transportation management perspective.

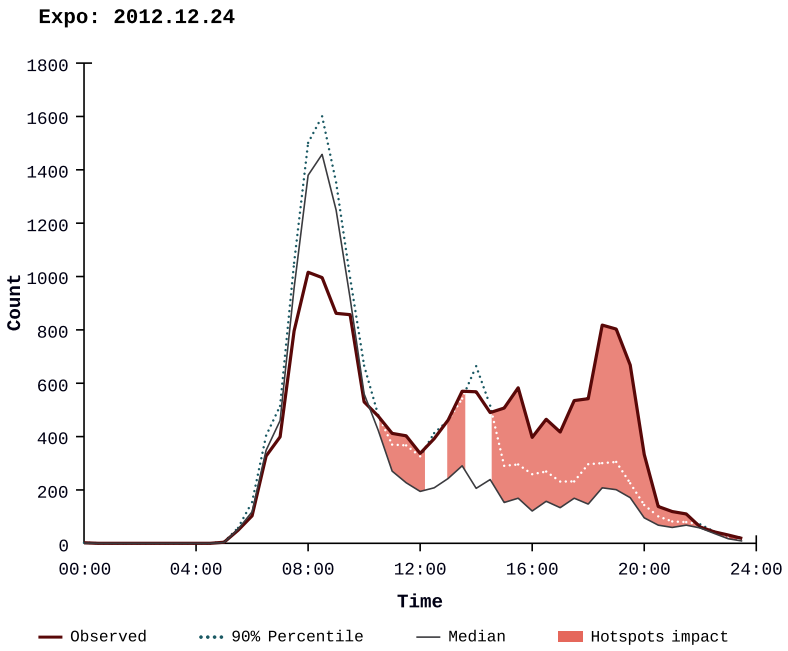


Figure 6.1: Example of the detection and measurement overcrowding hotspots using real data from the Singapore's EZLink system for the Expo area.

Table 6.1: General statistics of the study areas: averages (± std. dev.) and totals.

Area	Average daily arrivals	Avg. daily events	Number of hotspots	Avg. hotspot impacts
Stadium	4120.4 (± 1015.9)	0.2 (± 0.5)	103	382.9 (± 680.0)
Expo	14797.5 (± 5851.3)	2.5 (± 2.0)	70	2836.7 (± 4846.3)
Esplanade	4788.7 (± 930.5)	17.0 (± 6.4)	102	231.6 (± 430.8)

The dataset used consists of 4 months of smartcard public transport data from Singapore's EZLink system. This is a tap-in/tap-out system both for buses and subway (MRT), which means that we can infer both departure and arrival locations for any trip. For the purposes of this specific study, we selected trips that start/end in 3 areas that are sensitive to multiple special events: Stadium, Expo and Esplanade. The Stadium area has two major venues: the Singapore Indoor Stadium, which is mostly home to music concerts and sports events, and the Kallang Theatre, which is a 1680-seat auditorium that usually hosts live theater performances, operas and other cultural shows. Both venues are then capable of hosting events of various types and with different target audiences. By having two co-located venues, this area allows us to understand the effect of concurrent events in close-by venues. As for the Singapore Expo, it does not have any other significant venues on the vicinity but it has a large area of 123.000m^2 with several exhibition halls. Hence, it regularly hosts multiple events at the same time (usually large exhibitions and conventions), thus making this area far more challenging to analyze. The Esplanade area has more than 50 venues and is a lively touristic area near the business district. It has several shopping malls nearby and sits in front of the iconic marina bay of Singapore. Table 6.1 shows some descriptive statistics of these areas.

Table 6.2: General statistics on the data mined from the internet.

Source	Num. events study areas	Number of categories	Text description size (± std. dev.)	Retrieval type
eventful.com	1221	28	1112.3 (± 1337.1)	API
singaporeexpo.com.sg	58	28	124.9 (± 159.5)	scrapper
last.fm	11	-	901.2 (± 1037.5)	API
timeoutsingapore.com	568	49	411.8 (± 866.6)	scraper

6.3 Retrieving potential causes from the web

For each overcrowding hotspot we want to find a set of candidate explanations from the web. For this, we mainly take advantage of user-contributed event directories. Two general techniques exist to capture this crowd-generated data automatically, namely, application programming interfaces (APIs) and screen scraping. The choice entirely depends on the website. Some websites provide an exhaustive API that we can use to retrieve the data, while for others we need to resort to individual calls, page by page (screen scraping). Either way, access may be restricted or prohibited by terms of service. Therefore, we implemented individual event data retrievers for each website whenever it is so permitted. We use 5 different websites: eventful.com, upcoming.org, last.fm, timeoutsingapore.com and singaporeexpo.com.sg. For potential duplicates that share the same venue/area and day, we use the Jaro-Winkler string distance (Winkler, 1990) with a conservative threshold (e.g. $> 85\%$ similarity) to identify and merge them. Whenever we find different textual descriptions, we concatenate them.

Each event record contains title, venue, web source, date, start time, end time, latitude, longitude, address, url, description, categories, and when available the event price. Unfortunately, this information also contains plenty of noise. For example, the majority of start and end times are absent or "default" (e.g. from 00:00 to 23:59), and the same sometimes happens with latitude/longitude (e.g. center of the map). The latter can be corrected by using the venue name, but for the former, we could not determine any particular times. As a consequence, each such event is potentially associated to any impact hotspot of the corresponding day and area.

The description texts are run through a topic model, namely latent Dirichlet allocation (see Section 5.2), in order to represent each description as a distribution over topics. One key parameter for this process is the number of topics K. After trying with a range of values, from 15 to 40, the value that yielded the best model results was 25. We will assume this value for the remainder of this chapter. The α prior was set to $1/K$. With the purpose of understanding whether this was a safe choice, we ran several iterations with different initial values for α and they generally converged to similar outcomes.

For each event, we also capture two online popularity indicators, namely the number of Facebook likes and the number of hits in Google of the event title query. We retrieve the Facebook page with a semi-automatic procedure: we follow the event's url (which is sometimes a Facebook page) in search of candidate pages. Whenever there is more than one candidate, we manually select the correct one. For Google hits, we search with the event title within and without quotes (yielding

two separate features). Table 6.2 summarizes a few statistics of this dataset.

We can see that the most comprehensive sources are eventful and timeout, while the one with more detailed descriptions is eventful. Expo's homepage and last.fm have much less, yet very directed, information. The former contains all events that happen in Expo (thus a relevant filter in itself) while the latter is only focused on music events.

6.4 Proposed model

The individual event contributions to the hotspots are not observed, i.e. they are latent, but we do know that they contribute to the global observed impact. We will also assume that individual impacts are mutually exclusive (e.g. no one attends two events) and independently distributed, and that there will be a parcel that is unexplainable, i.e., some trips will neither be related to the extracted events nor to the usual commuting patterns. Thus, we say that the n^{th} hotspot impact, h_n, is given by

$$h_n = a_n + b_n + \epsilon, \tag{6.1}$$

where $\epsilon \sim \mathcal{N}(\epsilon|0, v)$ is the observation noise, a_n is the non-explainable component and b_n is the explainable one. The latter is itself a summation of the E_n events, $\{e_n^i\}_{i=1}^{E_n}$. Formally, we define a_n and b_n as

$$a_n \sim \mathcal{N}(a_n|\boldsymbol{\eta}_a^T \mathbf{x}_n^a, \beta_a) \tag{6.2}$$

$$b_n = \sum_{i=1}^{E_n} e_n^i, \text{ with } e_n^i \sim \mathcal{N}(e_n^i|\boldsymbol{\eta}_e^T \mathbf{x}_n^{e_i}, \beta_e), \tag{6.3}$$

where $\mathbf{x}_n^a$, $\boldsymbol{\eta}_a$, and β_a are the attributes, parameter vector, and variance, respectively, for the non-explainable component a_n.

The explainable part b_n, is determined by a sum of event contributions e_n^i. Each $\mathbf{x}_n^{e_i}$ corresponds to the individual attributes of the i^{th} event (e.g. topic-assignments, categories, Facebook likes, etc.) associated with the n^{th} observation, and $\boldsymbol{\eta}_e$ and β_e correspond to, respectively, the event attributes' parameters and the variance associated with the events' components. Notice that we assumed a linear-Gaussian model for the non-explainable and individual event contributions. This linear formulation will be kept for this chapter, and we will leave the extension to non-linear ones for Chapter 7. Figure 6.2 shows a representation of the proposed model as a probabilistic graphical model.

Our main goal is to estimate the values of a_n and e_n^i, so that they sum up to h_n. This relationship can be represented through the joint probability distribution

$$p(h_n, a_n, \mathbf{e}_n|\boldsymbol{\eta}_a, \boldsymbol{\eta}_e, \mathbf{X}_n) = p(h_n|a_n, \mathbf{e}_n)\, p(a_n|\boldsymbol{\eta}_a, \mathbf{x}_n^a) \left(\prod_{i=1}^{E_n} p(e_n^i|\boldsymbol{\eta}_e, \mathbf{x}_n^{e_i}) \right), \tag{6.4}$$

where we defined $\mathbf{e}_n = \{e_n^1, \ldots, e_n^{E_n}\}$ and $\mathbf{X}_n = \{\mathbf{x}_n^a, \mathbf{x}_n^{e_1}, \ldots, \mathbf{x}_n^{E_n}\}$ for compactness. It may be helpful to notice the relationship between Figure 6.2 and the expansion on the right-hand side of the equation, where we can see the conditional dependences.

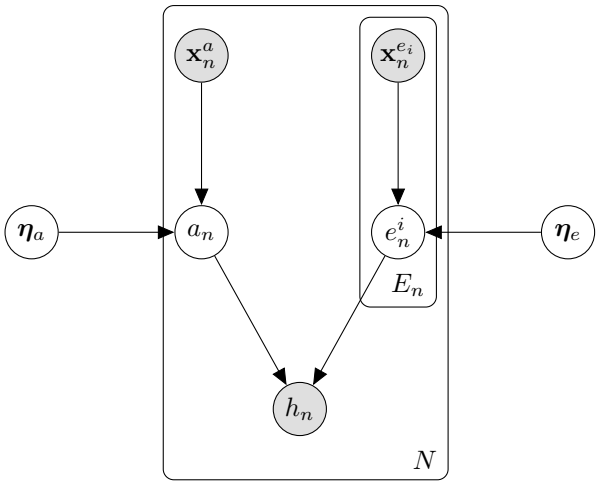


Figure 6.2: Graphical representation of the proposed model.

In order to simplify the notation, let $\boldsymbol{\eta} = (\boldsymbol{\eta}_a; \boldsymbol{\eta}_e)$. Given a dataset $\mathcal{D} = \{h_n, \mathbf{X}_n\}_{n=1}^N$ and assuming a zero-mean Gaussian prior $p(\boldsymbol{\eta})$ on the regression coefficients, we can estimate the posterior over $\boldsymbol{\eta}$ by making use of Bayes' rule, giving

$$p(\boldsymbol{\eta}|\mathcal{D}) \propto p(\boldsymbol{\eta}) \prod_{n=1}^{N} \int p(h_n, a_n, \mathbf{e}_n|\boldsymbol{\eta}, \mathbf{X}_n)\, da_n\, d\mathbf{e}_n. \tag{6.5}$$

Using the posterior distribution $p(\boldsymbol{\eta}|\mathcal{D})$ estimated using the entire dataset, we can compute the posterior distributions of the latent non-explainable component a_n, for each individual observation n, by again making use of Bayes' rule to give

$$p(a_n|h_n, \mathbf{X}_n) \propto \int p(h_n, a_n, \mathbf{e}_n|\boldsymbol{\eta}, \mathbf{X}_n)\, p(\boldsymbol{\eta}|\mathcal{D})\, d\boldsymbol{\eta}\, d\mathbf{e}_n, \tag{6.6}$$

and similarly for the explainable component $\mathbf{e}_n$.

The proposed model was implemented in the Infer.NET framework[2] (Minka et al., 2012). Infer.NET allows us to specify a probabilistic model programmatically, by exploring the concept of probabilistic programming. The main idea of probabilistic programming is to extend common programming languages by introducing random variables, which are extensions of standard types that can represent uncertain values. Hence, instead of a single value, each random variable represents a set or range of possible values, and has an associated distribution that assigns a probability to each possible value. The Infer.NET framework then provides the necessary approximate Bayesian inference tools that allow us to compute posterior distributions efficiently.

Notice how, without further assumptions, (6.5) and (6.6) have exact analytical solutions, which can be obtained by making use of the formulas for the conditional and marginal distributions of Gaussians in (B.6) and (B.7). In practice, Infer.NET allows us to constrain variables to be positive, which we do for the explainable and non-explainable components, a_n and $\mathbf{e}_n$, respectively. By doing so, exact inference is no longer possible. Hence, we instruct Infer.NET to perform approximate Bayesian inference using expectation propagation (EP) (Minka, 2001).

[2]The Infer.NET implementation of the proposed model can be found in: `http://amilab.dei.uc.pt/fmpr/why-so-many-people/`.

Area	CorrCoef	MAE	RRSE	R^2
Stadium	0.99	410.3	0.21	0.96
Expo	0.89	145.0	0.45	0.80
Esplanade	0.85	708.1	0.56	0.69

Table 6.3: Results for synthetic data.

6.5 Experiments

As it happens in many other cases (e.g. aggregated cell-phone statistics), we have access to total values but not to the individual contributions. This makes validating the proposed model a much harder task. Hence, we proceed by first testing it as if we had observed the individual contributions. We do this by generating simulated data that complies with our additive assumption. Only afterwards we test how well the proposed model fits with respect to the real total observed values.

Synthesized data experiments

If we cluster the events dataset from Section 6.3 using the events' characteristics, we end up with sets of events that are somehow related. Let's assume that each cluster centroid is assigned its own impact, manually or randomly. This value represents the impact of a hypothetical event, that does not necessarily exist in the database. Now, let us assign impacts to the real events using the distance to their cluster centroid, c. For each event e^k, its impact is determined by $dist(e^k, c)^{-1}$.

With this procedure, we are not forcing the structure of the proposed model into the data, i.e. we are not assigning specific parameter values to the coefficients $\boldsymbol{\eta}_a$ and $\boldsymbol{\eta}_e$, and using those pre-defined parameters to sample observations according to (6.2) and (6.3). In fact, we do not even know if there exist such parameters, $\boldsymbol{\eta}_a$ and $\boldsymbol{\eta}_e$, which are able to fit the simulated values. Instead, we use similarity between events to introduce consistency, regardless of area or day.

The individual impacts of simultaneously occurring events are added up and the resulting sum is perturbed according to some percentage of Gaussian noise, $\mathcal{N}(0, 0.1 \times b_n)$. The final result is provided to the proposed model as the observed hotspot impact. The estimated individual impacts are then compared to the ground truth (simulated) values according to four error statistics: the correlation coefficient (CorrCoef) gives an insight on how the results of the proposed model are correlated with the ideal results; the mean absolute error (MAE) provides the absolute magnitude of the error for each impact; the root relative squared error (RRSE) shows the quality of the model relative to a naive predictor based on the average; and the coefficient of determination (R^2) indicates how well the data fits the proposed model. Table 6.3 shows the results for the Stadium, Expo and Esplanade areas.

As the obtained results show, the proposed model has different performances throughout the different areas. In Stadium, it is able to replicate particularly well the contributions, which is not surprising since this area is more homogeneous than the others (often with only one event in a day). Despite being much more heterogeneous, in both Expo and Esplanade, the model can still achieve a significant correlation coefficient and considerably outperform the average-based predictor.

Area	CorrCoef	MAE	RRSE	R^2
Stadium	0.68	271.7	0.55	0.70
Expo	0.84	2002.7	0.69	0.52
Esplanade	0.41	192.6	0.84	0.29

Table 6.4: Results for real data from Singapore's EZLink system.

Real data experiments

Our observations consist of total hotspot impacts according to the definition in Section 6.2. In this section, we test the capability of the proposed model to recover such aggregated impacts without knowing the individual impacts. Hence, the model can only rely on the observed features such as location, day of week, event type, topics, etc., as represented by the vectors $\mathbf{x}_n^a$ and $\{\mathbf{x}_n^{e_i}\}_{i=1}^{E_n}$. This is done by first inferring the posterior distribution of the coefficients $\boldsymbol{\eta}_a$ and $\boldsymbol{\eta}_e$ with a subset of the observations (trainset) and then estimating the aggregated hotspot impacts for the remaining subset (testset). We apply 10-fold cross-validation (Bishop, 2006) and report the same error metrics as in the previous section. Table 6.4 shows a summary of the results.

Since hotspots can span through many consecutive hours, very large arrival totals can occur, particularly in the Expo and Esplanade areas. Thus, the relevance of MAE is difficult to assess. On the other hand, for these areas, the values for the correlation coefficient, RRSE and R^2 indicate that the model is able to provide good performance, while for the Esplanade the results are less conclusive.

Regardless of the fact that this is a more complicated task, the proposed model is able to approximate the totals well in two of the cases (Stadium and Expo). If the assumptions of the proposed model were wrong, the predictions should be considerably off, because the magnitude of the totals varies according to the time duration of the hotspot and because the individual event proportions could be wrong. The specific Esplanade case will be analyzed in the following section.

6.6 Explaining hotspots

The ultimate goal of the proposed model is to break down each overcrowding hotspot into a set of explanatory components. In this section, we present the results for the entire dataset. Before, we have validated individual component predictions through a synthetic dataset and the aggregated totals with the observations. This time, however, we do not observe the contributions of individual events. Even if we had access to individual participation data (e.g. through ticket sale statistics), it would not necessarily reveal the correct numbers of public transport users for that specific event. Thus, our evaluation will now be qualitative.

Figures 6.3, 6.4 and 6.5 illustrate some of the results.[3] For each hotspot, we show the global impact (inner circle) and the breakdown (outer circle). The area size of the inner circle is relative to the maximum hotspot impact observed in that location in the dataset. The outer circle will contain as many segments as potential explanatory events plus the non-explainable component (in red). For example, on

[3]Full set available in http://amilab.dei.uc.pt/fmpr/why-so-many-people/.

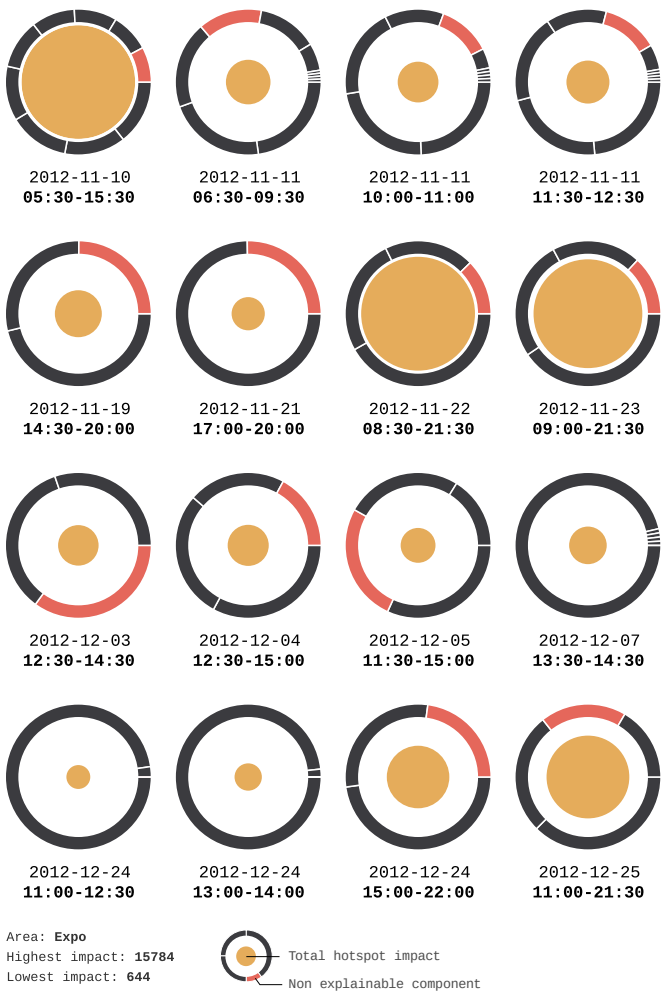


Figure 6.3: Breakdown of the total arrivals for 12 events from the Expo area.

November 10, 2012, Expo had a high impact hotspot (top left diagram in Figure 6.3) comprised of eight different events, with roughly the same size. The non-explainable component was in fact small (red segment). Differently, on November 19, 2012, the same area had 2 events, one of which explains almost half of a relatively small hotspot, if comparing with the previous case.

For Stadium and Expo, we can see that the non-explainable component is generally smaller than the explainable one and that the breakdown is not evenly distributed. This happens because the model maximizes consistency across different events. For example, two similar events in two occasions will tend to have similar impacts although the overall totals and sets of concurrent events may be different.

Cases with multiple hotspots in the same day are interesting to analyse. For example, in Figure 6.3, Expo had 3 hotspots on November 11, 2012, with minor fluctuations on the impacts and individual breakdowns. There were 10 different medium sized events (3 sale events, 2 movie and anime industry events, 1 parenthood and 1 pet ownership event, 2 home furniture and decoration related exhibits) that spanned throughout the day. Differently, in Stadium (Figure 6.4), the hotspots for February 22, 2013, have totally opposite behaviors. This was a fanmeet event with a Korean music and TV celebrity, that started at 20:00 (we note that the largest impact is between 17:30 and 21:00). While the model is confident in the first

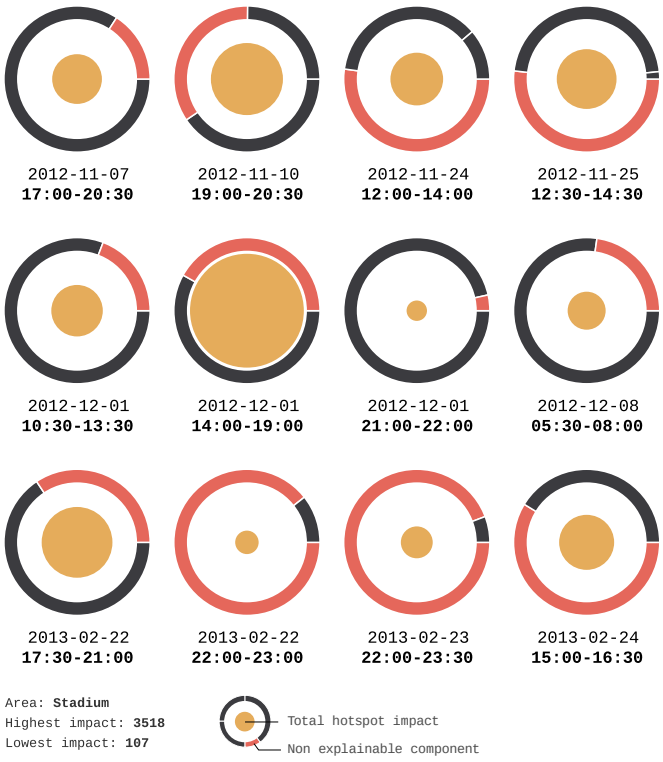


Figure 6.4: Breakdown of the total arrivals for 12 events from Stadium area.

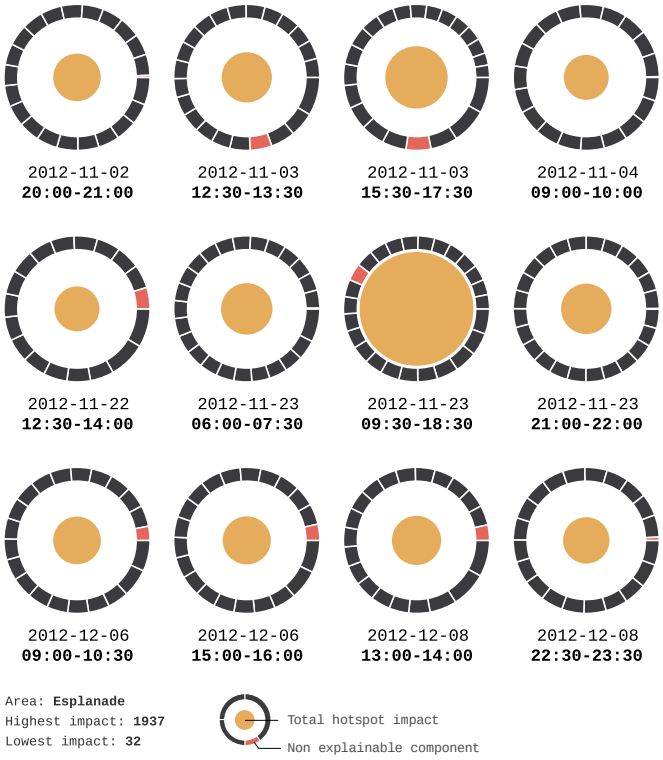
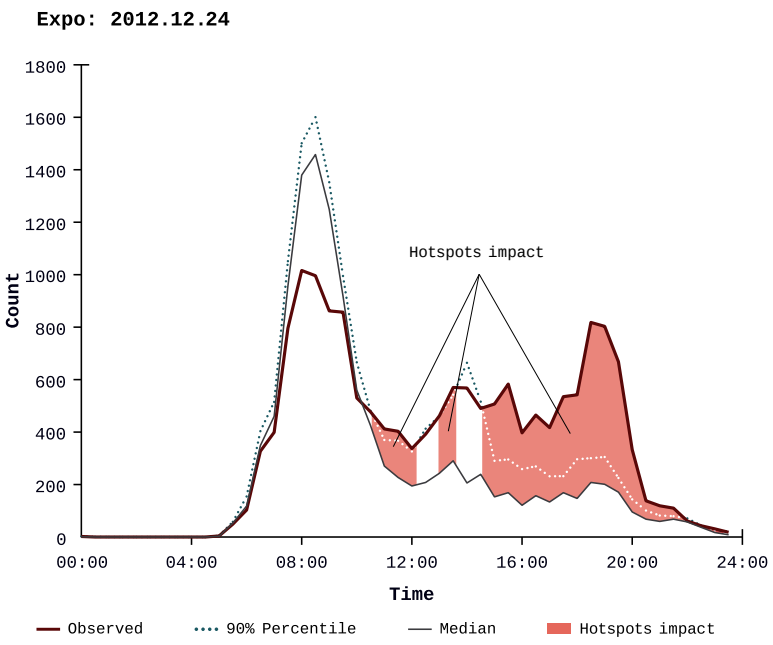


Figure 6.5: Breakdown of the total arrivals for 12 events from Esplanade area.



Event impacts:

Time	Megatex	kawin-kawin 2012	Non-explainable
11:00–12:30	644	0	0
13:00–14:00	847	14	0
15:00–22:00	2159	1326	1032

Figure 6.6: Impact breakdown for Expo on 24th of Dec. 2012 (same as Fig. 6.1).

hotspot, it does not assign the same explanation to the second one and leaves it mostly unexplained.

The case of Esplanade (Figure 6.5) shows unclear patterns as the proposed model was generally unable to go beyond an even breakdown. In fact, a careful look at the data shows that there are sometimes multiple small events being announced for that area, from game watching nights at bars to theatre sessions. Outliers do exist (e.g. opera concerts) but the algorithm would probably need more such cases to extract them. Nevertheless, it shows capability of ruling out some as insignificant events by assigning zero impact to them.

Let us now analyze a few cases in detail. In Figure 6.6, we show the hotspot breakdown of Figure 6.1 according to the proposed model. We notice that it was Christmas eve and there were two events: Megatex, an IT and electronics fair; Kawin-kawin makan-makan, a Malay products event. The model proposes that the majority of the impacts relate to the electronics event, which is intuitively plausible, particularly on the day before Christmas and knowing that Singapore has a well-known tech-savvy culture.

In Figure 6.7, we show the breakdown of a single hotspot, from 12:30 to 14:30 (the other 2 were filtered out due to small impact and duration). This was a tennis event, "Clash of Continents 2012", and people arrived mostly for the last final matches. The "Dance drama opera warriors" was held at 20:00 at the Kallang theatre. Intuitively, there is no surprise that an international sports event attracts more people than a classical music one. In fact, this is an example where the text descriptions play important roles. If the events were a pop concert (also music) and a local basketball game (also sports), the results could be drastically different.

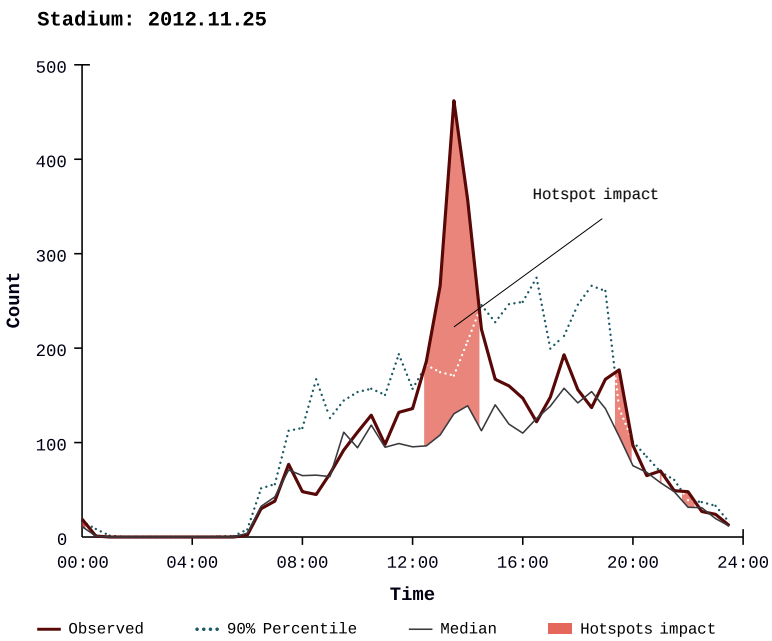
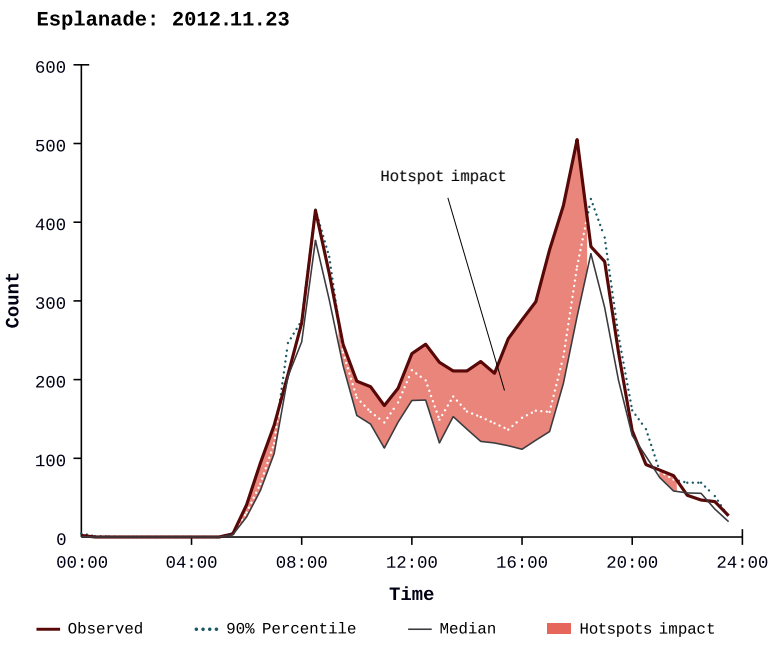


Figure 6.7: Impact breakdown for Stadium on the 25th of November 2012.

Finally, Figure 6.8 represents again the most challenging case for the proposed model — the Esplanade. Kalaa Utsavam is an Indian arts festival which has several events that, aggregated together, generate the largest impact. Intuitively, this is plausible given the presence of Indian-origin population and culture in Singapore. However, the results are not very clear. For example, "Ten years shooting home" is a photography contest event that may not have brought nearly as many people as the "International Conference on business management and information systems". Nevertheless, our analysis of the model and data suggests that a longer timeline and an improved data cleaning/filtering process should increase the quality of these results.

6.7 Conclusion

In this chapter, we presented a probabilistic model that breaks aggregated overcrowding hotspots into their constituent explanatory components. We extracted candidate explanations from the internet under the assumption that, except for habitual behavior (e.g. commuting), such crowds are often motivated by public events announced in the web. Since we do not observe the individual event's contributions, we treat them as latent variables and rely on the total sum and the event's features to constrain their estimation. The proposed model has an additive structure: the observed totals results from the sum of an explainable and a non-explainable component. The explainable component is further broken-down into the various candidate explanations retrieved from the internet.



Event impacts: 9:30h–18:30h

Quantity	Event title
128	Non-explainable
697	Kalaa Utsavam 2012
247	Kuo Pao Kun: A Life of Practice (theatre)
238	SLO – Mother Daughter Wife and Lover (opera)
145	SRT Holiday Programme – Fantastic Mr Fox (theatre)
128	Travel and Make Money Seminar
123	The Romance of Magno Rubio (theatre)
116	Ten Years of Shooting Home (photography)
115	Int. Conf. on Business Management and Information Systems

Figure 6.8: Impact breakdown for Esplanade on the 23th of November 2012.

This model was tested on a public transport dataset from the city-state of Singapore. We identified overcrowding hotspots by comparing the observed arrival counts (bus or subway arrivals) with a conservative threshold (90% quantile) at 30 minutes intervals. The hotspots were quantified by summing up consecutive excessive counts. For each such hotspot, we retrieved the potential explanations from several event announcement websites among other crowd-generated online sources, and we extracted relevant available information such as event title, category, venue, and description. We applied latent Dirichlet allocation (LDA) to extract topics from the text descriptions, which were then used to characterize the different events. All these features were organized together in the proposed model, which was implemented in the Infer.NET framework (Minka et al., 2012). Results with synthetic data show that the model is able to retrieve the correct results with a correlation coefficient (CorrCoef) of at least 85% and a coefficient of determination (R^2) higher than 0.85. The results with real data show that the same model is able to recover the observed total impacts with a correlation coefficient from 41.2% to 83.9% and an R^2 from 0.41 to 0.68, even though this is a harder task than what the model was built for. A qualitative analysis on a case study in Singapore shows that the results of the hotspot impacts' breakdowns into different possible explanations are intuitively plausible.

Chapter 7

Improving transportation demand prediction using crowds data

7.1 Introduction

In the previous chapter, we saw how a simple, yet effective, additive model could help us decompose an observed overcrowding hotspot into the contributions of various special events, such as sports games, concerts, operas, sales, demonstrations, festivals, etc. Motivated by the success of that work, in this chapter we apply the concept of additive models to the more general problem of predicting public transport usage in special event scenarios, by again correlating smartcard data records with context information mined from the Web. Hence, instead of just considering overcrowding hotspots, we now consider the entire time-series of public transport arrivals, which we model as a sum of a routine time-series component, that captures the routine behavior of a given place (e.g. commuting), and the contributions of a variable number of components that correspond to the events that occur in the neighborhood of that place. In doing so, we develop a general-purpose Bayesian additive framework, which, contrarily to typical approaches such as linear regression, neural networks or Gaussian process regression, possesses many interesting properties that make it particularly well suited for modeling transportation demand.

Unlike the model proposed in Chapter 6, in this chapter we propose a Bayesian additive model where the components are non-linear functions of their inputs, which we model as independent Gaussian processes (GPs) (Rasmussen and Williams, 2005). As our experiments show, by including additive components (GPs) that rely only on crowd-generated data regarding events, the proposed model is able to significantly improve the quality of the predictions. We derive an efficient approximate inference algorithm using expectation propagation (EP) (Minka, 2001), which, besides making predictions of the total number of public transport trip arrivals in a given place, it allows us to breakdown an observed time-series of arrivals into the contributions of the different components: routine commuting and individual special events. Figure 7.1 illustrates this application with actual results from the proposed model using real data from the Singapore's Indoor Stadium and Kallang Theatre, which share the same subway stop. On this day (November 25, 2012), the Indoor Stadium had a tennis tournament ("Clash of Continents") and the Kallang Theatre had an opera. Within the tennis tournament, there was a short Leona Lewis concert scheduled between two exhibition matches, sometime between 15:00 and 16:00. As

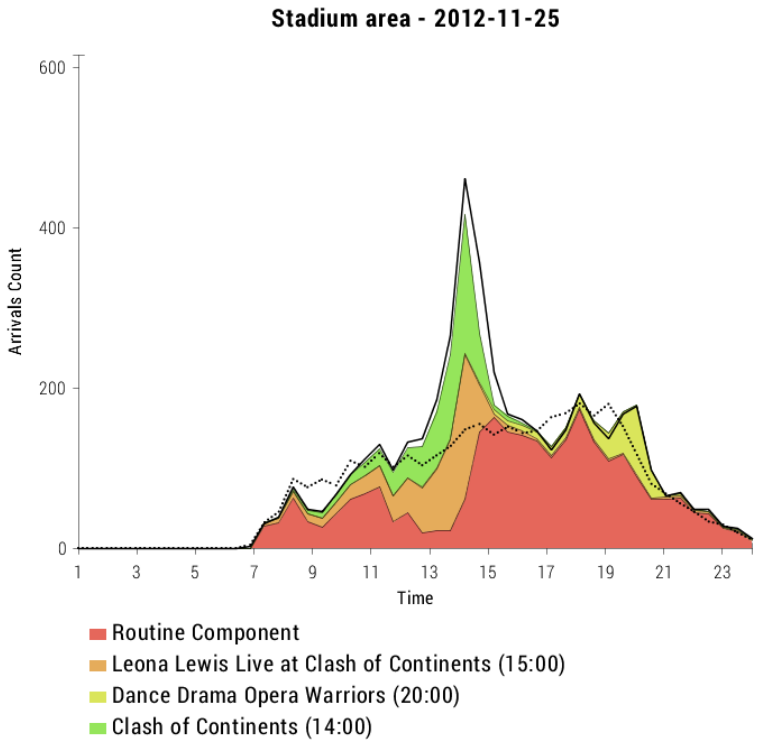


Figure 7.1: Breakdown of the observed time-series of subway arrivals (black solid line) into the routine commuting (area in red) and the contributions of events from the Singapore Indoor Stadium and Kallang Theatre (orange, yellow and green areas). The dotted line represents the median arrivals over all the days in the observed data that correspond the same weekday. Events start times are shown in parentheses.

in the previous chapter, the proposed model uses multiple features collected from the Web that will be described later, including start and end times, event venue, results from an automated web search, etc. In this example, it identifies that the large bulk of trips between 12:00 and 15:00 were arrivals to the Leona Lewis concert and to the tennis tournament. Then, after 17:00, there were arrivals to the opera (scheduled for 20:00) together with routine trips.

By applying the proposed framework to the problem of modeling public transport arrivals under special events scenarios, we are therefore able to (i) predict the distribution of the total number of arrivals that will be observed in the future considering all the events that are spatially and temporally close; (ii) disaggregate the time-series of arrivals into the contributions of a routine component and a variable number of event components, making predictions about the contribution of each future event separately. All this information can be of great value not only for public transport operators and planners, but also for event organizers and public transport users in general. Finally, by using a Bayesian approach, the proposed model can be easily adapted to perform online learning. Together with the efficient approximate inference algorithm developed, it has the ability to scale to very large datasets and to be deployed in practice.

Although we focus on a transportation application, it is important to note that this is a general-purpose methodology that can be extended to different application

domains, such as electrical signal disaggregation or source separation (Park and Choi, 2008). Indeed, the Bayesian additive framework described in this chapter can be of great value for any prediction task where knowing the importance (or contribution) of different inputs is required. For example, when modeling particle emissions, it is essential to have interpretable systems, so that researchers can understand how each of the individual factors (traffic, forest fires, kitchens, air conditioning/heating, industry) contributes to the total emissions values observed or forecasted. The same applies to transport management challenges, where operators and planners need to understand what originates demand fluctuations to mitigate them properly.

The remainder of this chapter is organized as follows. In the next section, we contextualize the proposed Bayesian additive model with Gaussian process components within the relevant literature. Section 7.3 provides motivation for the proposed Bayesian additive model, while the model itself is introduced and explained in Section 7.4. The corresponding experimental results are presented in Section 7.5. The chapter ends with the conclusions.

7.2 Additive models

Linear regression models provide an effective and attractively simple framework for understanding how each input variable relates with the observed target variables. However, they fail to capture non-linear dependencies between inputs and target variables, which are recurrent in the real-world. On the other hand, flexible models such as neural networks or Gaussian processes (GPs) lay on the opposite side of the spectrum, where the target variables are modeled as complex non-linear functions of all input variables simultaneously. Unfortunately, due to their black-box nature, the interpretativeness and the ability to understand how each input is contributing to the observed target are typically lost. Additive models (Hastie and Tibshirani, 1990) contrast with these by specifying the target variable to be the result of a linear combination of non-linear functions of the individual inputs. Due to this structured form, additive models provide an interesting tradeoff between interpretability and flexibility.

The typical approach in additive models is to rely on scatterplot smoothers for representing non-linear effects of the individual inputs in a flexible way (Hastie et al., 2003; Ravikumar et al., 2009). Additive models can then easily be fitted using a backfitting procedure (Hastie and Tibshirani, 1990), which iteratively fits each of the scatterplot smoothers to the residuals of the sum of the estimates of all the other smoothers, until a convergence criterion is met. The model proposed in this chapter contrasts with these works in several ways, particularly: (i) we consider the use of Gaussian processes instead of scatterplot smoothers; and (ii) we propose a fully Bayesian approach for inferring the posterior distribution of the individual function values using expectation propagation (Minka, 2001).

From the specific perspective of Gaussian processes, Duvenaud et al. (2011) proposed the additive GP: a GP model for functions that decompose into a sum of other low-dimensional functions. This is achieved through the development of a tractable kernel which allows additive interactions of all orders, ranging from univariate functions of the inputs to multivariate interactions between all input variables simultaneously. Although efficient in exploring all orders of interaction between inputs, additive GPs do not support a variable number of interacting functions as we

require for our practical application of public transport demand prediction, where there is a variable number of events happening. Furthermore, the Bayesian additive framework presented in this chapter is more flexible, in the sense that it allows to incorporate further restrictions on the models such as non-negativity constraints, as well as combining linear with non-linear functions or combining GPs with different covariance functions.

Compared to the ensemble learning literature, the proposed models shares several characteristics with Bayesian additive regression trees (BART) (Chipman et al., 2010). Namely, both approaches model the observations as a sum of non-linear functions. However, in BART these functions are regression trees that depend on all the input variables simultaneously. By imposing a prior that regularizes the fit by keeping the individual tree effects small, BART can be seen as combination of "weak learners", which are fitted using a Bayesian backfitting procedure based on Markov chain Monte Carlo methods. Hence, contrarily to the model proposed in this chapter, BART is not designed for generating interpretable models.

7.3 Problem formulation

Let y be the total number of public transport arrivals at a given time. Perhaps the most natural approach to model y is to consider it to be a function of time, the day of the week, whether or not it is a holiday, etc. We refer to these as routine features, $\mathbf{x}^r$, as they characterize the routine behavior of a given place. A wide majority of previous works focuses solely on these features (e.g. van Oort et al. (2015)). However, as previously discussed, there are several other dynamic aspects of transportation demand that need to be accounted for. Particularly, we are interested in the effect of special events. Let $\mathbf{x}^{e_i}$ be a feature vector characterizing a given event e_i, such as the venue, categories, tags, etc. Since the number of events that occur in a given area varies, we consider models of the form

$$y = f_r(\mathbf{x}^r) + \sum_{i=1}^{E} f_e(\mathbf{x}^{e_i}) + \epsilon, \tag{7.1}$$

where $\epsilon \sim \mathcal{N}(\epsilon|0, v)$ is the observation noise and E denotes the number of events that can affect the observed arrivals y. Hence, the number of events, E, varies between observations (although we assume it to be constant within each day). However, if we assume the functions, $f_r(\mathbf{x}^r)$ and $f_e(\mathbf{x}^{e_i})$, to be linear functions of their inputs, parameterized by a vector of coefficients $\boldsymbol{\eta}_r$ and $\boldsymbol{\eta}_e$ respectively, then we can write (7.1) as

$$y = (\boldsymbol{\eta}_r)^{\mathrm{T}}\mathbf{x}^r + (\boldsymbol{\eta}_e)^{\mathrm{T}}\left(\sum_{i=1}^{E}\mathbf{x}^{e_i}\right) + \epsilon = \boldsymbol{\eta}^{\mathrm{T}}\mathbf{x} + \epsilon, \tag{7.2}$$

where we defined $\mathbf{x} \triangleq (\mathbf{x}^r; \sum_{i=1}^{E}\mathbf{x}^{e_i})$ and $\boldsymbol{\eta} \triangleq (\boldsymbol{\eta}_r; \boldsymbol{\eta}_e)$. As we can see, in the case of linear functions and without further restrictions to the model (e.g. positivity constraints), the feature vectors of all events can be aggregated by summation, which reduces the problem to a simple linear regression. However, this allows the functions $f_r(\mathbf{x}^r)$ and $f_e(\mathbf{x}^{e_i})$ to have arbitrary values, which is not desired, since we know a

priori that the contributions of each component to the observed sum must be non-negative. Therefore, this formulation does not let us exploit our domain knowledge properly. Furthermore, for the particular application domain of transportation demand, the functions $f_r(\mathbf{x}^r)$ and $f_e(\mathbf{x}^{e_i})$ can be highly non-linear, as our experiments demonstrate (see Section 7.5). As soon as we start considering non-linear models, the distributive law used in (7.2) can no longer be applied, and the feature vectors of the different events $\mathbf{x}^{e_i}$ cannot be simply summed up. Moreover, the number of such vectors varies constantly from time to time. Aggregating these in order to allow a standard regression formulation, such as Gaussian process regression, to be applied is then a non-trivial research question. In the following section, we propose a Bayesian additive model that not only allows us to handle these issues but also leads to other attractive properties.

7.4 Bayesian additive model

7.4.1 Proposed model

The proposed Bayesian additive model builds on the assumption that there is a base routine component $y^r = f_r(\mathbf{x}^r)$ and a variable number of event components $y^{e_i} = f_e(\mathbf{x}^{e_i})$, whose contributions are summed up to obtain the total observed arrivals y in a given area. Since we want to constrain the values of the individual components, y^r and $\{y^{e_i}\}_{i=1}^E$, to be non-negative, we define the latter to be one-side truncated Gaussians[1], which we denote as

$$y^r \sim \mathbb{I}(y^r > 0)\,\mathcal{N}(y^r | f_r(\mathbf{x}^r), \beta_r), \tag{7.3}$$
$$y^{e_i} \sim \mathbb{I}(y^{e_i} > 0)\,\mathcal{N}(y^{e_i} | f_e(\mathbf{x}^{e_i}), \beta_e), \tag{7.4}$$

where $\mathbb{I}(a > 0)$ is an indicator function that takes the value 1 if and only if $a > 0$, β_r and β_e are variances of the routine and events components, respectively. Usually, one can assume that $\beta_r < \beta_e$ since the routine component is expected to be less noisy. The observed totals y are then defined to be Gaussian distributed

$$y \sim \mathcal{N}\Big(y \Big| y^r + \textstyle\sum_{i=1}^E y^{e_i}, v\Big). \tag{7.5}$$

Having specified the additive structure of the model, the next step is to specify how to model the individual components f_r and f_e. In this chapter, we use Gaussian processes (GPs) (Rasmussen and Williams, 2005) although the additive framework described above is general enough to allow a large variety of models to be applied. Nevertheless, for the sake of comparison and with the purpose of understanding certain aspects of the additive framework, we also provide a version with linear models for the components in Appendix C.4.

Letting the vectors $\mathbf{f}^r$ and $\mathbf{f}^e$ denote the functions $f_r(\mathbf{x}^r)$ and $f_e(\mathbf{x}^{e_i})$ evaluated for all feature vectors $\mathbf{x}^r$ and $\mathbf{x}^{e_i}$ respectively, we proceed by placing a GP prior on $\mathbf{f}^r$ and $\mathbf{f}^e$, such that $\mathbf{f}^r \sim \mathcal{GP}(m_r(\mathbf{x}^r) \equiv 0, k_r(\mathbf{x}^r, \mathbf{x}^{r'}))$ and $\mathbf{f}^e \sim \mathcal{GP}(m_e(\mathbf{x}^e) \equiv 0, k_e(\mathbf{x}^e, \mathbf{x}^{e'}))$, where for the sake of simplicity (and without loss of generality), we assumed the

[1] In principle, any non-negative distribution, e.g. negative binomial or Poisson, can be used. We chose the truncated Gaussian distribution for the sake of simplicity and because our preliminary experiments showed it to be a good fit.

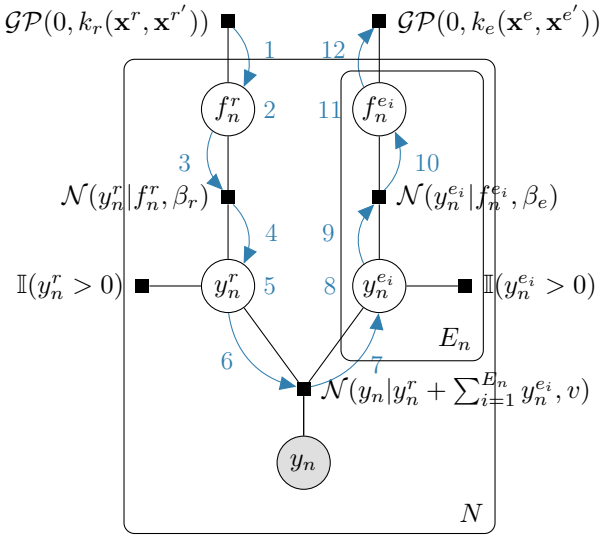


Figure 7.2: Factor graph of the proposed Bayesian additive model with Gaussian process components. The outer plate represents the observations, while the inner plate represents the events associated with each observation. The blue arrows represent the message-passing algorithm for performing approximate Bayesian inference. The second flow of messages starting from the GP factor for the events component that goes in the opposite direction is not shown.

GPs to have zero mean, so that the GPs are completely defined in terms of the covariance functions k_r and k_e. The generative process of the proposed Bayesian additive model can then be summarized as follows:

1. Draw $\mathbf{f}^r | \mathbf{X}^r \sim \mathcal{GP}(0, k_r(\mathbf{x}^r, \mathbf{x}^{r'}))$
2. Draw $\mathbf{f}^e | \mathbf{X}^e \sim \mathcal{GP}(0, k_e(\mathbf{x}^e, \mathbf{x}^{e'}))$
3. For each observation $n \in \{1, ..., N\}$
   (a) Draw routine component $y_n^r | f_r(\mathbf{x}_n^r), \beta_r \sim \mathbb{I}(y_n^r > 0)\, \mathcal{N}(y_n^r | f_r(\mathbf{x}_n^r), \beta_r)$
   (b) For each event e_i, with $i \in \{1, ..., E_n\}$
      i. Draw event contribution $y_n^{e_i} | f_e(\mathbf{x}_n^{e_i}), \beta_e \sim \mathbb{I}(y_n^{e_i} > 0)\, \mathcal{N}(y_n^{e_i} | f_e(\mathbf{x}_n^{e_i}), \beta_e)$
   (c) Draw total observed arrivals $y_n | y_n^r, \{y_n^{e_i}\}_{i=1}^{E_n} \sim \mathcal{N}(y_n | y_n^r + \sum_{i=1}^{E_n} y_n^{e_i}, v)$

where E_n denotes the number of events that are associated with the n^{th} observation. Figure 7.2 shows a factor graph representation of the proposed model, which will be particularly useful in the following section for deriving a message passing algorithm to perform approximate Bayesian inference using expectation propagation (EP) (Minka, 2001).

7.4.2 Approximate inference

Let $\mathcal{D}$ be a dataset of N observations, each one corresponding to the total number of arrivals (transportation demand) associated with the bus or subway stations that serve a certain special events' area in a given time interval. Formally,

$\mathcal{D} = \{\mathbf{x}_n^r, \mathbf{X}_n^e, y_n\}_{n=1}^N$, with $\mathbf{X}_n^e = \{\mathbf{x}_n^{e_i}\}_{i=1}^{E_n}$, and $\mathbf{x}_n^r$, $\mathbf{x}_n^{e_i}$ being the attributes of the routine and events components of observation n, respectively.

Given a dataset $\mathcal{D}$, our goal is two-fold: (i) compute the marginal distributions of the individual components y_n^r and $y_n^{e_i}$ and (ii) make predictions for new input vectors $\{\mathbf{x}_*^r, \mathbf{X}_*^e\}$. According to the factor graph in Figure 7.2, the joint distribution of the proposed model is given by

$$p(\mathbf{f}^r, \mathbf{f}^e, \mathbf{y}^r, \mathbf{Y}^e, \mathbf{y} | \{\mathbf{x}_n^r, \mathbf{X}_n^e\}_{n=1}^N) = \mathcal{N}(\mathbf{f}^r | \mathbf{0}, \mathbf{K}^r)\, \mathcal{N}(\mathbf{f}^e | \mathbf{0}, \mathbf{K}^e) \prod_{n=1}^N \mathbb{I}(y_n^r > 0)\, \mathcal{N}(y_n^r | f_n^r, \beta_r)$$
$$\times \left(\prod_{i=1}^{E_n} \mathbb{I}(y_n^{e_i} > 0)\, \mathcal{N}(y_n^{e_i} | f_n^{e_i}, \beta_e) \right) \mathcal{N}\left(y_n \middle| y_n^r + \sum_{i=1}^{E_n} y_n^{e_i}, v \right),$$

where we defined $\mathbf{y} \triangleq \{y_n\}_{n=1}^N$, $\mathbf{y}^r \triangleq \{y_n^r\}_{n=1}^N$ and $\mathbf{Y}^e \triangleq \{\mathbf{y}_n^e\}_{n=1}^N$, with $\mathbf{y}_n^e \triangleq \{y_n^{e_i}\}_{i=1}^{E_n}$. The covariance matrices $\mathbf{K}^r$ and $\mathbf{K}^e$ are obtained by evaluating the covariance functions $k_r(\mathbf{x}^r, \mathbf{x}^{r'})$ and $k_r(\mathbf{x}^e, \mathbf{x}^{e'})$, respectively, between every pair of inputs.

Unfortunately, the non-Gaussian truncation terms, $\mathbb{I}(y_n^r > 0)$ and $\mathbb{I}(y_n^{e_i} > 0)$, deem exact Bayesian inference to be computationally intractable. Hence, we proceed by developing a message-passing algorithm using EP in order to perform approximate Bayesian inference in the proposed model. In EP, the marginals $p(y_n^r)$ and $p(y_n^{e_i})$ are approximated via moment matching, thus resulting in the Gaussian distributions $q(y_n^r)$ and $q(y_n^{e_i})$ with the same mean and variances as $p(y_n^r)$ and $p(y_n^{e_i})$. EP is therefore able to approximate the non-Gaussian factors by local Gaussian approximations. However, these approximations are made in the context of all the remaining factors, which gives EP the ability to make approximations that are more accurate in regions of high posterior probability (Murphy, 2012).

Let the message sent from factor f to variable x be $m_{f\to x}(x)$. Similarly, let $m_{x\to f}(x)$ be the message sent from variable x to factor f. We can obtain a message-passing viewpoint of EP by defining the following update equations (Murphy, 2012):

- Messages from factors integrate out all variables except the receiving one

$$m_{f\to x}(x) = \int f(x, \mathbf{z}) \prod_{z\in\mathbf{z}} m_{z\to f}(z)\, d\mathbf{z}, \tag{7.6}$$

  with the integral being replaced by summation for all the z's in $\mathbf{z}$ which are discrete instead of continuous random variables.

- Messages from variables are the product of all incoming messages except that from the receiving factor

$$m_{x\to f}(x) = \prod_{h\in H_x\backslash\{f\}} m_{h\to x}(x), \tag{7.7}$$

  where $h \in H_x\backslash\{f\}$ is used to denote all factors h in the neighborhood of x, H_x, except the factor f.

- Marginals are the product of all incoming messages from neighbor factors

$$q(x) = \text{proj}\left[\prod_{f\in F_x} m_{f\to x}(x) \right], \tag{7.8}$$

where F_x denotes the set of factors in the neighborhood of x and the projection operation, $\text{proj}[p(x)]$, corresponds to finding the approximate distribution $q(x)$ that matches the moments of $p(x)$. Notice how this updates are the same as in exact inference with belief propagation (Bishop, 2006; Koller and Friedman, 2009) except for the projection operation.

Making use of this viewpoint of EP, we derive a message-passing algorithm for performing approximate Bayesian inference in the proposed model. This algorithm consists of 12 steps as illustrated in Figure 7.2. The blue arrows represent the directionality of the message flow and the blue labels denote the step number. Although not represented, there is a second flow of messages starting from the GP factor for the event components that goes in the opposite direction of the one depicted. In practice, these two flows of messages in opposite directions are implemented in parallel for added efficiency. Also, as the figure suggests, all the messages correspond to 1-dimensional Gaussians, which allows them to be represented compactly. A detailed derivation of all the steps of message-passing algorithm is provided in Appendix C.3. As previously mentioned, a version of the proposed Bayesian additive model with linear components instead of GPs is provided in Appendix C.4. In this case, the message-passing algorithm requires a smaller number of steps.

Two key advantages of the proposed framework are then its general applicability and extensibility. In fact, the building blocks for many interesting extensions have already been laid down through the proposed model. For instance, one could extend the model to account for effects of weather or seasonality in the observed arrivals. This could simply be done by including seasonality features in the routine component, but we could go a step further and introduce a new separate GP component, as this would allow us to estimate the effect of seasonality in the observed transportation demand. In fact, the equations for the new messages would be similar to the ones for the routine component, although in this case we would not wish to constrain the marginals to take only non-negative values. Therefore, there is a wide variety of interesting applications that could be developed just by making small adaptations to the proposed model and its inference algorithm in Appendix C.3.

7.4.3 Predictions

In the previous subsection, we discussed how to compute the posterior distribution of the latent variables given the observed totals using expectation propagation. This allows us to understand how transportation demand breaks down as a sum of a routine component and the contributions of the various events that take place in the neighborhood of a given bus or subway station. This, by itself, is of great value for public transport operators, urban planners and event organizers. However, we also want to make predictions for the "shares" of upcoming events and, ultimately, for the total estimated demand.

Let $\mathbf{x}_*^r$ be the features of the routine component for a given time and date, and let $\mathbf{X}_*^e = \{\mathbf{x}_n^{e_i}\}_{i=1}^{E_n}$ be the set of feature vectors characterizing the events that will take place. The EP algorithm in Appendix C.3 provides us with approximate posterior distributions for $\mathbf{f}^r$ and $\mathbf{f}^e$ given by $q(\mathbf{f}^r) = \mathcal{N}(\mathbf{f}^r|\boldsymbol{\mu}^r, \boldsymbol{\Sigma}^r)$ and $q(\mathbf{f}^e) = \mathcal{N}(\mathbf{f}^e|\boldsymbol{\mu}^e, \boldsymbol{\Sigma}^e)$ (kindly see Step 1 in Appendix C.3). These estimates can be used to compute the predictive mean and variance of f_*^r and $\{f_*^{e_i}\}_{i=1}^{E_*}$, as in standard Gaussian process regression, classification and GPC-MA (see Eqs. 4.36 and 4.37). The predictive

Table 7.1: Descriptive statistics of the two study areas.

Area	Average daily arrivals ± std.	Average daily events ± std.	Maximum daily events	Num. days without events
Stadium	4101 (± 925)	0.230 (± 0.554)	3	114 (82.014%)
Expo	15027 (± 5515)	2.446 (± 1.986)	8	23 (16.547%)

mean and variance for f_*^r are then given by (Rasmussen and Williams, 2005)

$$\mathbb{E}_q[f_*^r|\mathbf{f}^r, \mathbf{x}_*^r, \{\mathbf{x}_n^r\}_{n=1}^N] = (\mathbf{k}_*^r)^{\mathrm{T}}(\mathbf{K}^r + \tilde{\mathbf{\Sigma}}^r)^{-1}\tilde{\boldsymbol{\mu}}^r \tag{7.9}$$

$$\mathbb{V}_q[f_*^r|\mathbf{f}^r, \mathbf{x}_*^r, \{\mathbf{x}_n^r\}_{n=1}^N] = k_r(\mathbf{x}_*^r, \mathbf{x}_*^r) - (\mathbf{k}_*^r)^{\mathrm{T}}(\mathbf{K}^r + \tilde{\mathbf{\Sigma}}^r)^{-1}\mathbf{k}_*^r, \tag{7.10}$$

and similarly for the events variables $\{f_*^{e_i}\}_{i=1}^{E_*}$. We can then use the predictive mean and variance for f_*^r to estimate the share of the routine component as

$$\begin{aligned} p(y_*^r|\mathbf{f}^r, \mathbf{x}_*^r, \{\mathbf{x}_n^r\}_{n=1}^N) &= \mathbb{I}(y_*^r > 0) \int \mathcal{N}(y_*^r|f_*^r, \beta_r)\, p(f_*^r|\mathbf{f}^r, \mathbf{x}_*^r, \{\mathbf{x}_n^r\}_{n=1}^N)\, df_*^r \\ &\approx \mathcal{N}(y_*^r|\mu_*^r, v_*^r). \end{aligned} \tag{7.11}$$

This approximation is again made by moment matching (a derivation of these moments is provided in Appendix C.5), yielding

$$\mu_*^r = \mathbb{E}_q[f_*^r] + \sqrt{\mathbb{V}_q[f_*^r]}\frac{\mathcal{N}(z_*^r)}{\Phi(z_*^r)}, \tag{7.12}$$

$$v_*^r = \mathbb{V}_q[f_*^r]\left(1 - z_*^r\frac{\mathcal{N}(z_*^r)}{\Phi(z_*^r)} - \left(\frac{\mathcal{N}(z_*^r)}{\Phi(z_*^r)}\right)^2\right), \tag{7.13}$$

where $z_*^r \triangleq \mathbb{E}_q[f_*^r]/\sqrt{\mathbb{V}_q[f_*^r]}$ and $\Phi(\cdot)$ is the Gaussian cumulative distribution function. As for the equations for estimating the number of arrivals that will be caused by a given event $y_*^{e_i}$, they are analogous to the ones presented above for the number of routine arrivals y_*^r.

Finally, the predictive posterior distribution for the transportation demand (total number of arrivals) is given by

$$\begin{aligned} p(y_*|\mathbf{x}_*^r, \mathbf{X}_*^e, \mathcal{D}) &= \int \mathcal{N}\left(y_*\middle|y_*^r + \sum_{i=1}^{E_*} y_*^{e_i}, v\right)\mathcal{N}(y_*^r|\mu_*^r, v_*^r) \\ &\quad \times \prod_{i=1}^{E_*}\mathcal{N}(y_*^{e_i}|\mu_*^{e_i}, v_*^{e_i})\, dy_*^r\, dy_*^{e_1}\cdots dy_*^{e_{E_*}} \qquad (7.14) \\ &= \mathcal{N}\left(y_*\middle|\mu_*^r + \sum_{i=1}^{E_*}\mu_*^{e_i}, v + v_*^r + \sum_{i=1}^{E_*} v_*^{e_i}\right). \qquad (7.15) \end{aligned}$$

7.5 Experiments

The proposed Bayesian additive model with Gaussian process components (BAM-GP) was implemented in the Julia programming language[2] and evaluated in the

[2]Source code is available at: http://amilab.dei.uc.pt/fmpr/

Table 7.2: Five examples of the topics extracted by LDA.

Topic 7	Topic 8	Topic 11	Topic 21	Topic 24
congress	food	home	sale	music
event	bazaar	design	book	rock
confer	fiesta	interior	adida	song
annual	hotel	live	shop	michael
mta	wine	luxuri	john	learn
health	revolut	furnitur	deal	love
fpso	modul	hous	bag	john
week	analyt	renov	robinson	ford
world	event	idea	beauti	live
servic	restaur	furnish	warehous	download

context of the public transportation trip arrivals to event areas in Singapore. This dataset differs from the one used in Chapter 6, by considering one extra month of data: August 2013, and by only focusing on two study areas: the Stadium and the Expo. Each of these areas is served by its own subway station, whose number of arrivals we are trying to predict and dissect. Given this data, our goal is then two-fold:

- predict the total number of arrivals by half-hour in a given area in the presence of events;
- decompose the observed total of arrivals into the contributions of the routine component and the various events that took place in that area.

In order to achieve these goals, information about planned events is collected by mining the Web using the procedure described in Section 6.3. However, as we shall discuss in the following section, the data preparation procedure is now different.

Data preparation

As previously mentioned, we consider two study areas: the Stadium and the Expo. For the five months of public transport data, a dataset of events was retrieved from the Web, either through screen scrapping or, when available, through the direct use of APIs. Namely, we collected events information from the same sources as in Chapter 6: eventful.com, singaporeexpo.com.sg, upcoming.org, last.fm and timeoutsingapore.com. The duplicate event titles that also share the same venue and day were merged by again making use of the Jaro-Winkler string distance (Winkler, 1990). Table 7.1 provides some descriptive statistics of the collected data.

The events information consists of the title, venue, date, start time, end time, latitude, longitude, address, url, text description and categories/tags. From this information, we extract features such as the venue, whether the event has started/ended, the time to the event start/end, the event duration, if it is a multiday event or not, etc. Since the taxonomies of the different event sources vary significantly, the categories/tags provided became hard to include in a prediction model. Alternatively, we propose the use of a web search engine in order to characterize the events according to their subject. With this aim, we use the event titles and venue names

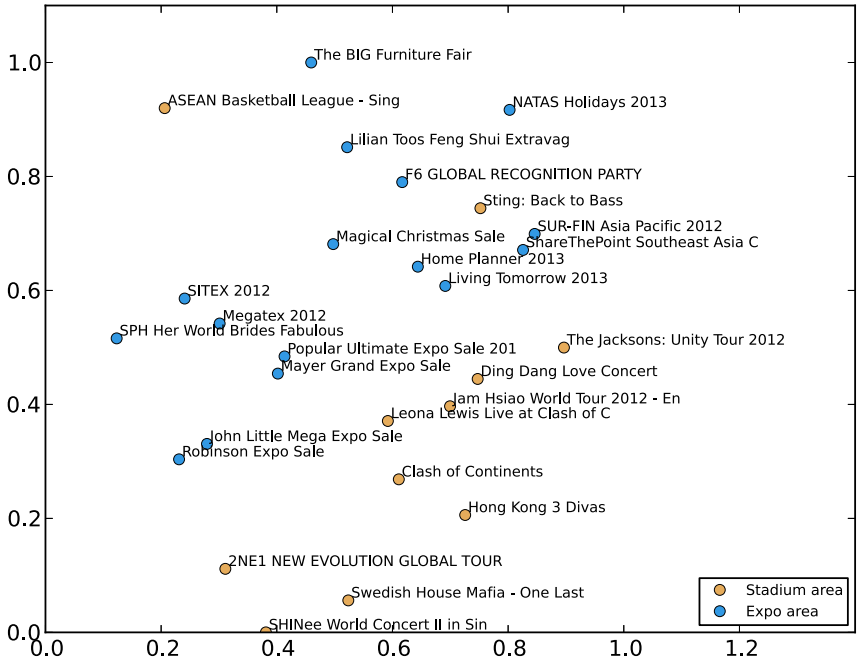


Figure 7.3: Visualization of the topic proportions for a sample of the events data using multidimensional scaling (MDS).

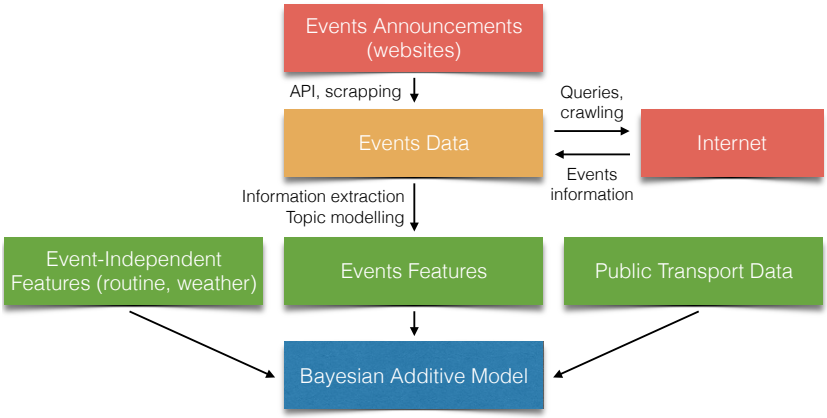


Figure 7.4: Flowchart of the data preparation process.

as queries and then we apply a latent Dirichlet allocation (LDA) topic model (Blei et al., 2003) to the obtained search results (titles and snippets together). This contrasts with the approach used in the previous chapter, where we applied LDA to the textual descriptions of the events directly. These descriptions are often very short and uninformative, and in practice, we found that using a web search engine over event titles and venues produces better results.

The number of topics in LDA was set to 25 based on an empirical analysis of the obtained topic distributions. Table 7.2 shows the top ten words for five example topics extracted by the LDA algorithm. The inferred topic distributions of the different events (in form of topic weights for each event) are then used as lower-dimensional representations of their search results.

Table 7.3: Results for estimating the total arrivals in the Stadium area using 10-fold cross-validation.

	Evaluation: all times			Evaluation: event periods only		
Model	CorrCoef	RAE	R^2	CorrCoef	RAE	R^2
Linear Reg. (routine only)	0.646	0.642	0.417	0.649	0.735	0.092
Linear Reg. (routine + events)	0.739	0.620	0.543	0.709	0.620	0.502
GP (routine only)	0.667	0.616	0.445	0.654	0.707	0.117
GP (routine + events)	0.777	0.567	0.603	0.751	0.581	0.564
BAM-LR	0.737	0.605	0.544	0.694	0.582	0.474
BAM-GP	**0.795**	**0.556**	**0.632**	**0.811**	**0.503**	**0.658**

Table 7.4: Results for estimating the total arrivals the Expo area using 10-fold cross-validation.

	Evaluation: all times			Evaluation: event periods only		
Model	CorrCoef	RAE	R^2	CorrCoef	RAE	R^2
Linear Reg. (routine only)	0.581	0.723	0.338	0.390	0.816	0.098
Linear Reg. (routine + events)	0.707	0.617	0.500	0.557	0.743	0.300
GP (routine only)	0.718	0.576	0.514	0.621	0.670	0.341
GP (routine + events)	0.750	0.547	0.543	0.676	0.668	0.382
BAM-LR	0.661	0.652	0.436	0.484	0.772	0.229
BAM-GP	**0.796**	**0.472**	**0.633**	**0.736**	**0.565**	**0.540**

Figure 7.3 shows a 2-D visualization of the inferred topic proportions for a random sample of the events using multi-dimensional scaling (MDS) (Borg and Groenen, 2005), a technique which seeks to find low-dimensional representations of the data while preserving the original distances between the data points. As the figure evidences, events with similar characteristics tend to be in the same region of the space. For example, the two electronics and IT fairs, SITEX and Megatex, are near each other. Similarly, the John Little and the Robinson (two large department stores) sales also appear together. More generally, we can notice the majority of the music-related events (e.g. Swedish House Mafia, SHINee, 2NE1, Leona Lewis, Jam Hsiao, etc.) being in same region of the space, separated from the rest of the events. Our hypothesis is then that events with similar topic distributions share similar effects on the observed arrivals and also on the general mobility patterns of a given place.

As for the routine features, we use the weekday, time (discretized in half-hour bins) and holiday information. The overall process of retrieval, information extraction and modeling of the events data is summarized in Figure 7.4.

Since some of the extracted features, especially some of the inferred topics, can turn out to be redundant for the arrivals prediction task and may actually decrease performance of the prediction algorithms, a simple feature selection procedure was used. It consists of a greedy search algorithm that starts with an empty set of features and then iteratively adds to this set the feature that yields the best improvement for a simple linear regression model evaluated using 10-fold cross-validation. This way, the feature selection process is kept efficient and completely independent from the proposed models. Using this algorithm, we were able to discard part of

the topics that were less likely related with the impact of events on urban mobility and that could in fact affect negatively the performance of all the prediction models. Indeed, a manual inspection of the prediction errors for the Stadium area revealed that the biggest improvement with the use of events' information was related to the discrimination between sports events and different types of concerts (e.g. with different music styles). In this case, the feature selection algorithm correctly chose topics that could help to discriminate between these different event types. At the same time, the majority of the time- and venue-related features were kept by the algorithm.

Arrivals prediction

The proposed model is evaluated using 10-fold cross validation, where the observations are ordered by time. Furthermore, the samples that belong to the same day are treated as a whole, so that they are assigned either to the test or train set altogether. This ensures that the model is only provided with information that is available in practice (recall that our goal with the prediction model is to make predictions far ahead of time, so that public transport operators are able to make changes accordingly). Since the proposed Bayesian additive models have two extra parameters, the variances of the routine and the events components (β_r and β_e), an additional 80/20 split is made on the train set of each fold in order to obtain a separate validation set for optimizing the values of β_r and β_e. For the sake of simplicity, a grid search procedure is used to set these parameters. The proposed model (BAM-GP) is then compared with the following baselines:

- two Bayesian linear regression models: one that uses only routine features, and another that corresponds to the model in Eq. 7.2, which uses both routine and event features;
- two Gaussian process models: one that considers event features and one that does not; in the case of the GP with information about events, the features of the multiple events that correspond to each observation are aggregated in the same way as with linear regression: by summing their values;
- and a version of the proposed Bayesian additive model that uses linear models for the routine and the events components (BAM-LR) as described in Appendix C.4. Notice how this approach is similar in spirit to the model proposed in Chapter 6. It differs mainly by considering a routine component and by using truncated Gaussians to model the values of the components.

The likelihood variance v and the strength of the prior over the coefficients of the linear regression models were set using the same grid search approach used for β_r and β_e. All the GPs, including the ones used in BAM-GP, use squared-exponential covariance functions (Rasmussen and Williams, 2005). The likelihood variance and length-scales of the GPs were determined by maximizing the marginal likelihood of the observations.

As measures of the quality of the predicted results, we report the following standard evaluation metrics: correlation coefficient (CorrCoef), relative absolute error (RAE) and coefficient of determination (R^2). Besides a global evaluation, we also provide error metrics only for the periods when events are about to start (time

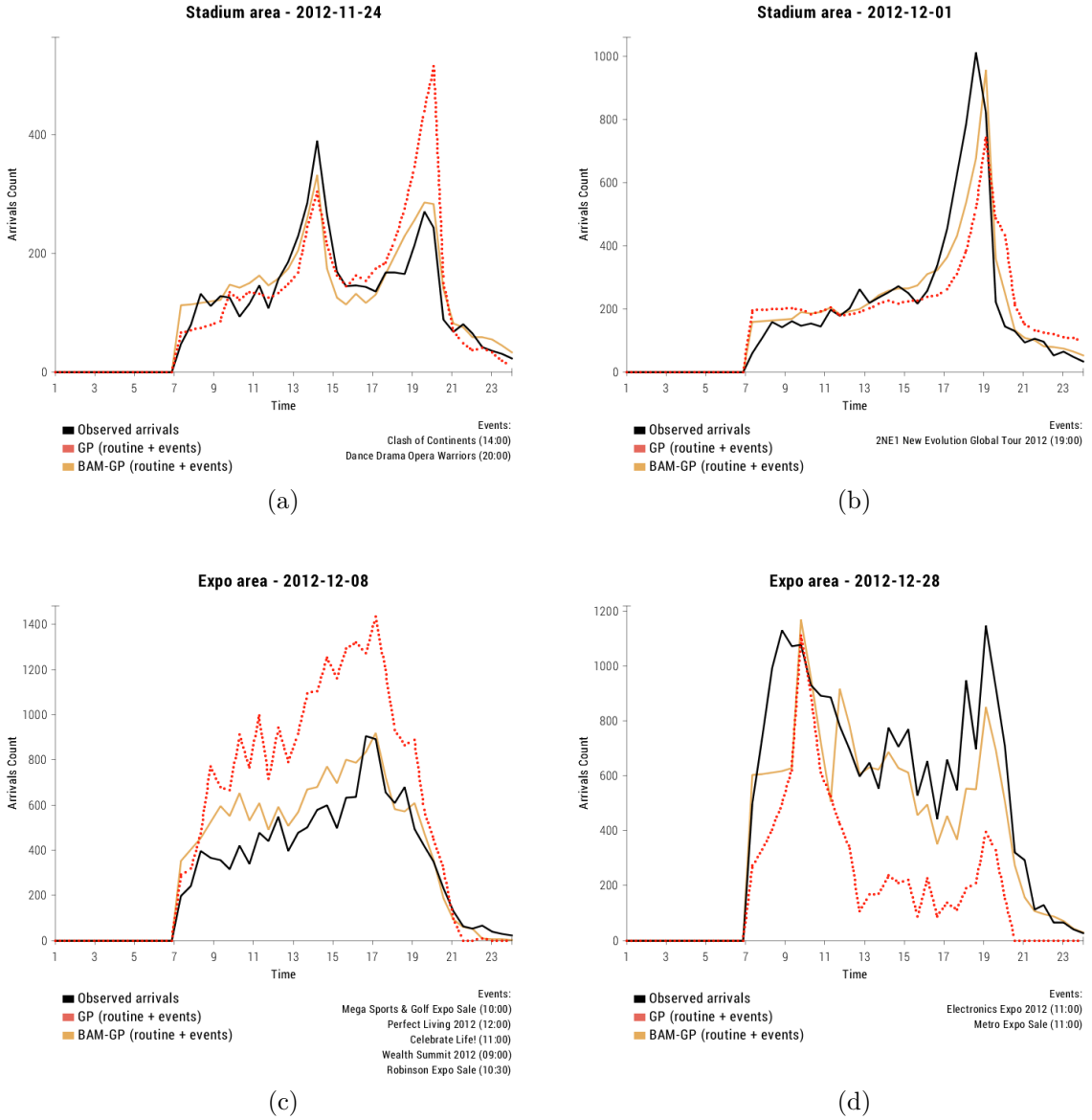


Figure 7.5: Comparison of the predictions of BAM-GP (orange solid line) with the true observed arrivals (black solid line) and the predictions of the GP model (red dotted line) on four example days. Events start times are shown in parentheses.

to event start is less than an hour) or ongoing, so that the contribution of the models that include event features can be more evident.

Tables 7.3 and 7.4 show the results obtained for the Stadium and Expo areas, respectively. As it can be seen, there is a clear advantage in including information regarding events in the public transport arrivals prediction models, which can lead to gains in R^2 of 42% and 23%, in the Stadium and Expo areas respectively, when compared to the best model that just uses routine information. If we focus only on the event periods, then this difference becomes even more significant, yielding gains of 462% and 54%, respectively. Not surprisingly, the improvements are higher in the Stadium area, since it has far less concurrent events (see Table 7.1), which makes the data considerably easier to understand and model. Also, as expected, the GP-based approaches compare favourably to their linear counterparts showing

a clear advantage especially in the Expo area, which supports our intuition that this is an highly non-linear problem.

The results in Tables 7.3 and 7.4 show that BAM-GP outperforms all the other baselines in both areas, where the GP model that uses routine and event features is the second best approach. Since in the Stadium area there are few simultaneous events, the feature aggregation problem described in Section 7.3 becomes less severe, and the difference between BAM-GP and the GP model becomes less significant when compared with the Expo area, which has an average of 2.446 daily events that generally overlap in time.

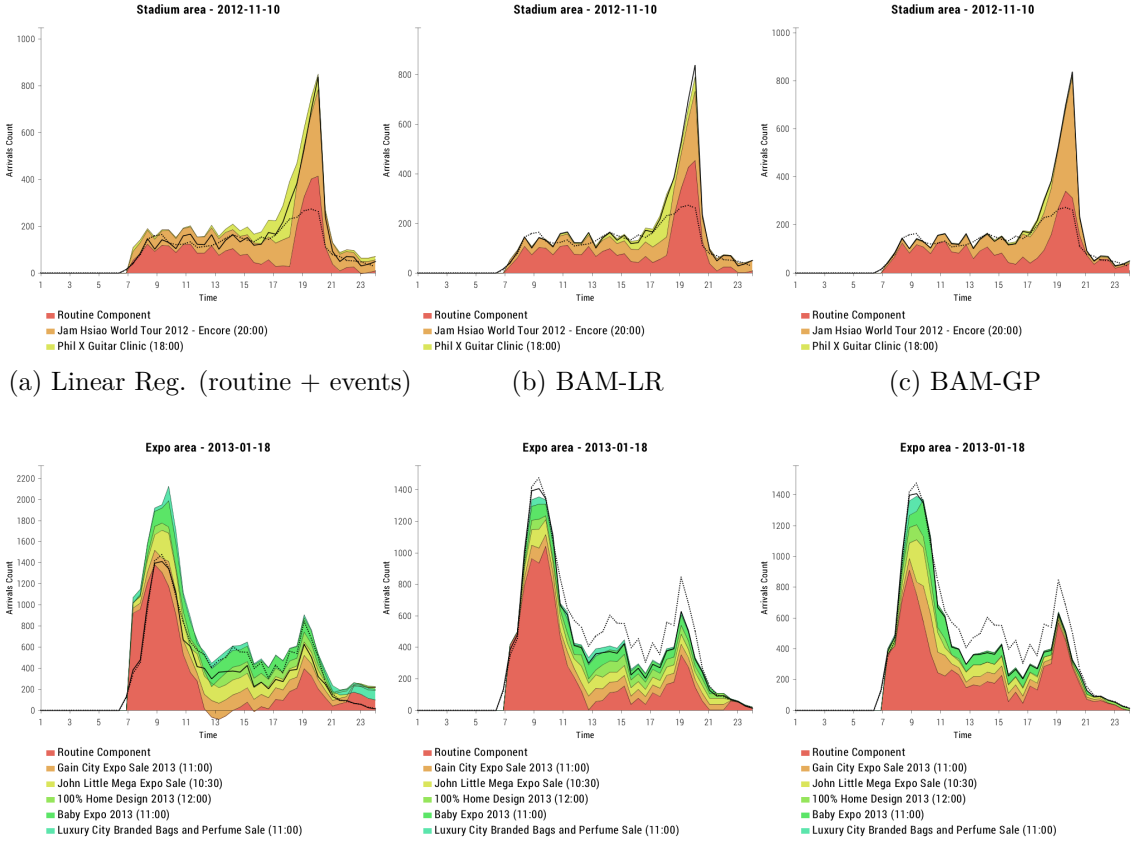
In order to illustrate some of these differences in their real-world context, we plotted the predictions of BAM-GP and GP models against the true observed arrivals for four example days in Figure 7.5. The obtained results highlight the practical implications of the improvements obtained by the proposed additive model. For example, in Figure 7.5a the GP model over-estimates the transportation demand around 19:00 by approximately 200 people, and it under-estimates the number of arrivals around the same time in Figure 7.5b, making an error of the same magnitude again. On the other hand, the estimates produced by BAM-GP are much closer to the observed number of arrivals. However, there is a phase shift of half an hour between the true and estimated peaks before the pop concert at 19:00. We hypothesize that this is due to the fact that 2NE1 is a very popular band in the southeast Asia and, therefore, people tend to arrive earlier to the show in order to guarantee a better spot. In fact, findings like these motivate us to explore the development of online popularity indicators to be included in the predictions models in our future work. Lastly, Figures 7.5c and 7.5d show two additional examples for the Expo area, where we also can see that BAM-GP provides much more accurate predictions than the GP model. These differences can be quite significant in terms of magnitude, thus making them likely to have major impacts in the transportation system.

Arrivals decomposition

In order to analyze decomposition results generated by the additive model, we need to take a closer look at the posterior marginal distributions on the routine component y^r and on the events components $\{y^{e_i}\}_{i=1}^{E}$ estimated by BAM-GP by running the EP inference algorithm on the entire dataset. Since it is impossible to obtain ground truth for this particular decomposition problem, our analysis will be qualitative rather than quantitative, and more based on common sense. With this aim, the performance of BAM-GP will be compared with the following baselines:

- the linear regression model from Eq. 7.2, where the inferred posterior distribution of the weights $\boldsymbol{\eta}$ is used to compute the posterior on the components y^r and $\{y^{e_i}\}_{i=1}^{E}$ for each individual observation by making use of Bayes theorem;
- and BAM-LR, where the EP algorithm described in Appendix C.4 is used to compute the marginal distributions of the routine and events components.

Figure 7.6 shows the results obtained by the three different approaches (columns) for two illustrative example days in the two study areas (rows). Let us start by analyzing the first row of examples, which correspond to the 10th of November 2012 in the Stadium area. From Figure 7.6a it can be seen that the component values



(a) Linear Reg. (routine + events) (b) BAM-LR (c) BAM-GP

(d) Linear Reg. (routine + events) (e) BAM-LR (f) BAM-GP

Figure 7.6: Results obtained by 3 different approaches (columns) on two examples days in two different areas (rows) for disaggregating the total observed arrivals (black solid line) into the contributions of the routine component and the various nearby events. The dotted line represents the median arrivals over all the days in the observed data that correspond to the same weekday. Events start times are shown in parentheses.

estimated by the linear regression model do not add up the total observed arrivals, which makes the output of this approach harder to use in practice. The BAM-LR decomposition from Figure 7.6b matches closer to the observed totals, however still it estimates arrivals caused by the Jam Hsiao concert long after that event is over, which is a naive mistake. Figure 7.6c shows that BAM-GP not only overcomes those problems, but it also provides more intuitive results. It assigns a significantly larger bulk of the demand to the big concert by Jam Hsiao (a young star mandopop singer widely popular in asian counties) that took place in the Indoor Stadium, as opposed to the small guitar clinic in the Kallang Theatre.

The results for the Expo area (Figure 7.6d) illustrate another weakness of the linear regression model. By not incorporating any constraints on the components, the estimated number of arrivals at 13:00 due to routine commuting becomes negative. Employing the truncated Gaussians for the components distributions, the Bayesian additive models do not suffer from this problem. However, as Figure 7.6e evidences, the simpler BAM-LR model is once again suffering from the problem of assigning a significant share of the arrivals to events when their are about to end

(around 21:00). The proposed non-linear model (BAM-GP) makes a much more reasonable estimate with that respect. This is a consequence of the fact that the relation between arrivals to an event and its end time, which is used as one of the model features, is non-linear. Moreover, it is expected that this relation would be also dependent on the type of event, which is something that a simple linear model cannot capture.

Finally, Figure 7.7 shows six additional illustrative decompositions produced by BAM-GP. All these examples further support the idea that BAM-GP is producing reasonable and well-informed disaggregations of the total observed arrivals into the contributions of routine commuting and the effects of the various events. For example, Figure 7.7c shows a case where the proposed additive model estimates a very small localized contribution for the event, which is not surprising because this was a small-sized event with a very narrow target audience that took place in a not so popular venue. Similarly, in Figures 7.7d and 7.7e the model is clearly assigning larger shares to the "Asia Pacific Food Expo" and "Megatex", which is reasonable since the former is a particularly large food festival in Singapore and the latter is a popular electronics and IT showcase. Another interesting example is shown in Figure 7.7f. On that day (January 19, 2013), the Singapore Expo had a concert at night by Sally Yeh, a cantopop singer and actress from Taiwan. As the figure depicts, the proposed model is assigning the majority of the late arrivals in this area to the concert.

7.6 Conclusion

In this chapter, we proposed BAM-GP — a Bayesian additive model (BAM) with Gaussian process (GP) components that allows for an observed variable to be modeled as a sum of a variable number of non-linear functions on subsets of the input variables. We developed an efficient approximate inference algorithm using expectation propagation (EP), which allows us to both make predictions about the unobserved totals and to estimate the marginal distributions of the additive components. The proposed model is then capable of being flexible, while retaining its interpretability characteristics. We apply BAM-GP to the problem of estimating public transport arrivals in special event scenarios. Using a five months dataset of Singapore's fare card system and crowd-generated data about special events mined from the Web, we show that the model presented not only outperforms others that do not account for information about events, thus verifying the value of internet-mined data produced by large crowds for understanding urban mobility, but also outperforms other more general models that do account for event information. Furthermore, due to its additive nature and Bayesian formulation, BAM-GP is capable of estimating the posterior marginal distributions that correspond to routine commuting and the contributions of the various events, which is of great value for both public transport operators/planners and event organizers. Finally, we believe that the presented methodology is quite general and that it can be easily adapted beyond the transportation domain such as, for example, in the analysis of financial time-series, cell-phone call records or electrical consumption signals.

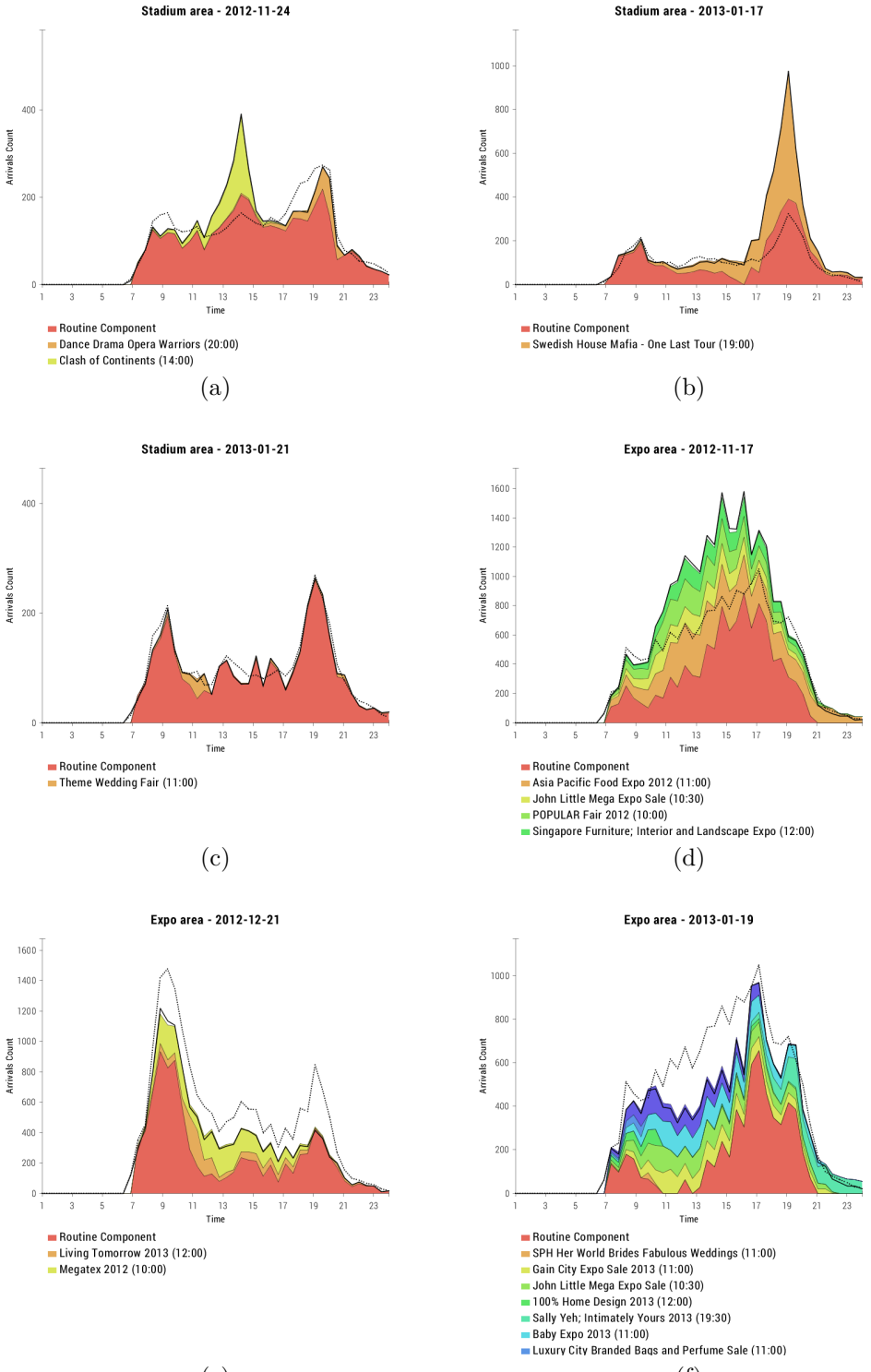


Figure 7.7: Results obtained by BAM-GP for disaggregating the total observed arrivals (black solid line) in 6 example days into the contributions of the routine component and the various nearby events. The dotted line represents the median arrivals over all the days in the observed data that correspond the same weekday. Events start times are shown in parentheses.

Chapter 8

Conclusions and future work

This thesis presented various probabilistic models for learning from crowdsourced data. As dataset sizes grow, crowdsourcing provides an attractive solution for efficiently labeling large volumes of data, especially due to online work-recruiting platforms such as Amazon mechanical turk (AMT). At the same time, as people share more and more information on the internet about what goes on in the real-world, the potential for solving complex machine learning problems and understanding certain real-world phenomena becomes unique. Unfortunately, the knowledge provided by crowds also brings many interesting challenges that must be addressed, like how to deal with the noise and uncertainty in these heterogenous information-sharing environments, how to cope with the fact that individuals are often biased and differ in terms of expertise, or how to develop machine learning approaches that make proper use of this data for building more accurate models of reality.

Making use of the framework of probabilistic graphical models, several solutions were proposed throughout this thesis in order to address some of these challenges. We began by considering the labels provided by multiple annotators as noisy replacements for the true outputs y in situations where these are hard or expensive to obtain, and developed several probabilistic models of increasing complexity for coping with the different levels of annotator expertise that are commonly encountered in practice when learning from crowds. First, a new class of models was introduced, in which the reliabilities of the different annotators are treated as latent variables. Within this class of latent expertise models, a classification approach based on logistic regression models was proposed (MA-LR). This approach was the base for developing MA-CRF — an extension of conditional random fields (CRFs) for learning from sequences labeled by multiple annotators. Using real data from AMT, we saw that the proposed model can lead to gains up to 22% in F1-score for named entity recognition tasks when compared to commonly used approaches such as majority voting.

Having clearly demonstrated the importance of annotator-aware models, we moved on to non-linear classification methods by developing a generalization of Gaussian process classifiers to multiple annotator settings — GPC-MA. By treating the unknown true labels as latent variables, this model is able to estimate the levels of expertise of the different annotators, thereby allowing it to compensate for their biases and to obtain better estimates of the ground truth labels. This was empirically validated using AMT data for a sentiment polarity and a music genre classification task, where the GPC-MA was shown to outperform traditional approaches by as

much as 0.178 in F1-score. Furthermore, an active learning methodology was developed, which allows us to select which instance should be labeled next and which annotator should label it. Using this methodology, the proposed model was shown to clearly improve over random selection approaches, while producing savings in annotation cost of more than 76%.

Motived by the success of the aforementioned approaches, and also realizing that the majority of the tasks for which crowdsourcing platforms are most popular belong to the fields of natural language processing and computer vision, two supervised topic models for learning from crowds were proposed. The fact that crowdsourcing is particularly popular within these research communities is easy to understand since the latter seek to mimic behaviors that are natural and easy for humans, such as understanding the meaning of a sentence or the content of an image, but that can be quite complex for machines due to the high dimensionality of the data. Hence, the amount of labeled data needed to compensate for this issue makes the use of crowdsourcing solutions very appealing. As it turns out, supervised topic models are particularly good for dealing with this type of data. Therefore, two supervised topic models were proposed for learning from complex high-dimensional data labeled by crowds: one for classification (MA-sLDAc) and another for regression tasks (MA-sLDAr). Using real data from AMT, MA-sLDAc was empirically shown to significantly outperform state-of-the-art approaches at classifying news articles and images according to their content. Similarly, MA-sLDAr demonstrated its superior predictive capabilities over other commonly-used approaches at predicting the ratings of movies and restaurants based on the text of the reviews.

In the second part of this thesis, the use of crowdsourced data as an additional input for improving machine learning models was considered. Focusing on the problem of modeling public transportation demand in the presence of special events such as sports games, concerts or festivals, a probabilistic model was proposed which uses internet-mined data to explain non-habitual overcrowding hotspots. Given real data from Singapore's public transport system and crowd-generated information about special events that was collected from the Web, the proposed model was shown to be able to predict the size of an overcrowding hotspot with a correlation coefficient ranging from 41.2% to 83.9% and an R^2 ranging from 0.41 to 0.68, depending on the area of study. But most importantly, a qualitative analysis of the results showed that the model is able to breakdown the excess demand into the shares of different possible explanations in an intuitively plausible manner.

Despite its linear assumptions, the overcrowding model was able to obtain very promising results. This inspired the development of a Bayesian additive model with Gaussian process components (BAM-GP) for the more complex task of predicting the time-series of public transport demand in the presence of special events. As with the overcrowding model, BAM-GP also relies on crowdsourced data mined from the Web for explaining the effect of events. However, by using non-linear models (GPs) for representing the relationship between the input features and the values of the individual additive components, the proposed model was empirically demonstrated to outperform state-of-the-art approaches by as much as 41% in R^2 during event periods based on 5 months of public transport data from Singapore. Furthermore, due to its additive nature and Bayesian formulation, BAM-GP was shown to be capable of estimating the posterior marginal distributions that correspond to routine commuting and the contributions of the various events, which is of great value for

both public transport operators/planners and event organizers.

In summary, this thesis proposed a collection of probabilistic models for learning from crowdsourced data. These were validated in various real-world applications, thereby demonstrating their value and also the wide applicability of crowdsourced data. Indeed, this type of data is more ubiquitous than it is sometimes realized. For example, in websites like IMDb.com, Yelp.com, Booking.com, AirBnB.com, Uber.com and Amazon.com different users provide ratings of movies, restaurants, hotels, accommodations, drivers and consumer products, respectively. However, these users have different reliabilities and some can even be spammers that give either the lowest or highest rating to all instances. Furthermore, they can have certain biases. For instance, a moviegoer might not be a fan of a particular genre and therefore rates all movies of that genre lower than they actually deserve. In fact, the same that happens with movies and genres happens with products and brands, restaurants and cuisines (e.g. Western, Mediterranean, Japanese, Thai, etc.) or with hotels and guest preferences (e.g. fast WiFi versus a large TV). Hence, future work could explore the application of the multiple-annotator models and ideas developed in this thesis to some of these problems, provided that these companies are willing to share their data. In fact, for many of these applications the proposed approaches could be extended to also model user tastes, such that the variables corresponding to the annotators' biases could be conditioned on their individual preferences.

Closely related to the idea of modeling users' individual tastes is another possible future work direction, which consists in relaxing the assumption that the annotators' expertise is independent of the instances that they are labeling. In situations where the task difficulty varies significantly between instances, it might be important to include such dependence in the annotator-aware models proposed in the first part of this thesis. For example, in a document classification task, this would allow the model to expect less reliable answers for documents that are next to the decision boundary between two classes, even if the annotators are generally reliable. Although conditioning the annotators' reliabilities on the instances that they are labeling would increase model complexity and make inference and learning harder, it could be especially advantageous in situations where different annotators are more qualified in certain regions of the input space. An example would be for the task of music genre classification. In this task, it is intuitive that annotators provide more reliable answers for songs that belong to the input space regions that correspond to the music styles that they know best. Hence, for songs with similar characteristics, it is expected that they provide equally reliable answers.

Another research line that justifies being explored in future work is related to the development of active learning strategies. In Section 4.5, we saw how active learning could be used to reduce annotation cost and to get the most out of a given budget, by cleverly choosing which instance to label next and who is the most qualified annotator to provide that label. However, although efficient and effective, the methodology proposed is suboptimal, since it considers each of these selection problems individually. Hence, the instance-annotator pair found might not necessarily be the best one if we consider both objectives together. In future work, approaches that jointly select the instances and the annotators could be explored.

Moving on to the use of crowdsourced data as additional input features, there are several ideas that deserve to be explored in future work. Perhaps the most obvious one relates to the application of the proposed models in other similar contexts, such

as traffic and telecommunications data. By making use of information mined from the Web and text mining techniques, it should be possible to understand disruptions in both road and telecommunications networks, and to forecast demand under the presence of events. Indeed, all the methodologies developed for modeling public transportation demand would be readily applicable to those problems. But the applicability of the proposed models can go even further. For example, the Bayesian additive framework developed in this thesis can be of great value for any prediction task where knowing the importance (or contribution) of different inputs is required, such as in source separation or electrical signal disaggregation, where the goal is to determine the power usage of individual home appliances given the whole-home electrical consumption data.

From a more general perspective, the proposed Bayesian additive models could be applied to any supervised machine learning problem where there is a need for interpretable models. As previously discussed, the proposed additive approaches, namely BAM-GP, provide an interesting tradeoff between simple linear models and more flexible models such as standards GPs. While fully interpretable, linear models are commonly overly simplistic. On the hand, more flexible models such as neural networks or GPs generally have a more black-box nature. Hence, although their predictions may be accurate, it is hard to understand what internally is leading the model to predict a specific result, which for some problems might be crucial. For example, when modeling air quality through the presence of pollutants, it is essential to have interpretable systems, so that researchers can understand how each individual factor (traffic, forest fires, kitchens, air conditioning/heating, industry, etc.) contributes to the total values observed or forecasted. Similarly, in computer-aided diagnosis (CAD), it is extremely important to have interpretable models in order to understand what drives the system to make a given diagnosis. Hence, in future work, we would like to explore variants of BAM-GP in a wide range of machine learning problems that could benefit from interpretable models and see how they compare to other state-of-the-art approaches.

One last future work direction that is worth mentioning is the relation between what crowds say online about a given event or its performer and the impact that it actually causes in transportation systems or urban mobility in general. Using information mined from the internet about events, we created simple Web-search-based popularity indicators which improved the proposed models. However, this remains a significant open question, involving aspects such as time-dependent dynamics, sentiment and social influence.

Appendix A

Probability distributions

A.1 Bernoulli distribution

The Bernoulli is a distribution over a binary variable $x \in \{0, 1\}$. The conjugate prior to the Bernoulli is the beta distribution. Its parameter is constrained by $\mu \in [0, 1]$.

$$\text{Bernoulli}(x|\mu) = \mu^x(1-\mu)^{1-x} \tag{A.1}$$
$$\mathbb{E}[x] = \mu \tag{A.2}$$
$$\mathbb{V}[x] = \mu(1-\mu) \tag{A.3}$$

A.2 Beta distribution

The beta is a distribution over a continuous variable $x \in [0, 1]$. It is the conjugate prior to the parameters of a Bernoulli. Its parameters α and β are constrained by $\alpha > 0$ and $\beta > 0$.

$$\text{Beta}(x|\alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\, x^{\alpha-1}(1-x)^{\beta-1} \tag{A.4}$$
$$\mathbb{E}[x] = \frac{\alpha}{\alpha+\beta} \tag{A.5}$$
$$\mathbb{V}[x] = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)} \tag{A.6}$$

Here $\Gamma(\cdot)$ is the gamma function, defined as $\Gamma(x) = \int_0^\infty u^{x-1}e^{-u}\, du$.

A.3 Dirichlet distribution

The Dirichlet is a distribution over K random variables $x_k \in [0, 1]$. It is the multivariate generalization of the beta distribution. The Dirichlet is the conjugate prior to the parameters of a multinomial distribution. Its parameters α_k are constrained by $\alpha_k > 0$.

$$\text{Dirichlet}(\mathbf{x}|\boldsymbol{\alpha}) = \frac{\Gamma\big(\sum_{k=1}^K \alpha_k\big)}{\prod_{k=1}^K \Gamma(\alpha_k)} \prod_{k=1}^K x_k^{\alpha_k - 1} \tag{A.7}$$

$$\mathbb{E}[x_k] = \frac{\alpha_k}{\sum_{j=1}^{K} \alpha_j} \tag{A.8}$$

$$\mathbb{V}[x] = \frac{\alpha_k\big(\sum_{j=1}^{K} \alpha_j - \alpha_k\big)}{\big(\sum_{j=1}^{K} \alpha_j\big)^2\big(\sum_{j=1}^{K} \alpha_j + 1\big)} \tag{A.9}$$

$$\mathbb{E}[\log x_k] = \Psi(\alpha_k) - \Psi\left(\sum_{j=1}^{K} \alpha_j\right) \tag{A.10}$$

Here $\Psi(\cdot)$ is the digamma function, defined as

$$\Psi(x) = \frac{d \log \Gamma(x)}{dx}. \tag{A.11}$$

A.4 Gaussian distribution

The univariate Gaussian is a distribution over a continuous variable $x \in \Re$. Its parameters μ and σ^2 are constrained by $\mu \in \Re$ and $\sigma^2 > 0$.

$$\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right) \tag{A.12}$$

$$\mathbb{E}[x] = \mu \tag{A.13}$$

$$\mathbb{V}[x] = \sigma^2 \tag{A.14}$$

The conjugate prior for the mean μ is another Gaussian, while the conjugate prior to the inverse variance $\rho = 1/\sigma^2$, called the precision, is the gamma distribution. The conjugate prior for both μ and ρ is known as the Gaussian-gamma distribution.

Its multivariate extension for a D-dimensional vector $\mathbf{x}$ is given by

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{\det(\boldsymbol{\Sigma})^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\intercal}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right) \tag{A.15}$$

$$\mathbb{E}[\mathbf{x}] = \boldsymbol{\mu} \tag{A.16}$$

$$\mathrm{cov}[\mathbf{x}] = \boldsymbol{\Sigma} \tag{A.17}$$

In this case, the conjugate prior for the inverse covariance matrix $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$, the precision matrix, is the Wishart. The conjugate prior for both $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$ is then the Gaussian-Wishart.

A.5 Multinomial distribution

Following part of the literature on machine learning, namely on topic models, in this thesis we refer to a multinomial distribution when we actually mean a "categorical" or "discrete" distribution. Notice that the multinomial is the multivariate generalization of the binomial, and hence it is a distribution over counts. In this thesis, we use the term "multinomial", to refer to a multinomial where the number of observations is 1.

In the context of this thesis, the multinomial is a distribution over a K-dimensional binary variable x, such that $x_k \in \{0, 1\}$ and $\sum_k x_k = 1$. The conjugate prior to

the multinomial is the Dirichlet distribution. Its parameters are constrained by $\mu_k \in [0, 1]$ and $\sum_k \mu_k = 1$.

$$\text{Multinomial}(x|\boldsymbol{\mu}) = \prod_{k=1}^{K} \mu_k^{x_k} \tag{A.18}$$

$$\mathbb{E}[x_k] = \mu_k \tag{A.19}$$

$$\mathbb{V}[x_k] = \mu_k(1 - \mu_k) \tag{A.20}$$

A.6 Uniform distribution

The uniform is a simple distribution over a continuous variable x, such that $x \in [a, b]$ and $b > a$.

$$\text{Uniform}(x|a, b) = \frac{1}{b - a} \tag{A.21}$$

$$\mathbb{E}[x] = \frac{b + a}{2} \tag{A.22}$$

$$\mathbb{V}[x] = \frac{(b - a)^2}{12} \tag{A.23}$$

Appendix B

Gaussian identities

B.1 Product and division

Given two (multivariate) Gaussian distributions $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, the product is given by

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) = Z^{-1}\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{B.1}$$

where

$$\begin{aligned}\boldsymbol{\mu} &= \boldsymbol{\Sigma}(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2)\\ \boldsymbol{\Sigma} &= (\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1})^{-1}.\end{aligned}$$

The normalization constant is given by

$$\begin{aligned}Z^{-1} &= (2\pi)^{-D/2}|\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2|^{-1/2}\exp\Big(-\frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\Big)\\ &= \sqrt{\frac{|\boldsymbol{\Sigma}|}{(2\pi)^D|\boldsymbol{\Sigma}_1||\boldsymbol{\Sigma}_2|}}\exp\left(-\frac{1}{2}(\boldsymbol{\mu}_1^T\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2^T\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2 - \boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu})\right).\end{aligned} \tag{B.2}$$

Similarly, for division we have

$$\frac{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)}{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)} = Z^{-1}\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{B.3}$$

where

$$\begin{aligned}\boldsymbol{\mu} &= \boldsymbol{\Sigma}(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2)\\ \boldsymbol{\Sigma} &= (\boldsymbol{\Sigma}_1^{-1} - \boldsymbol{\Sigma}_2^{-1})^{-1}.\end{aligned}$$

The normalization constant is given by

$$Z^{-1} = \sqrt{\frac{|\boldsymbol{\Sigma}||\boldsymbol{\Sigma}_2|}{(2\pi)^D|\boldsymbol{\Sigma}_1|}}\exp\left(-\frac{1}{2}(\boldsymbol{\mu}_1^T\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2^T\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2 - \boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu})\right) \tag{B.4}$$

B.2 Marginal and conditional distributions

On the other hand, if we have a joint Gaussian distribution $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with prevision matrix $\boldsymbol{\Lambda} \triangleq \boldsymbol{\Sigma}^{-1}$ and we define the following partitions

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{pmatrix}, \qquad \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{pmatrix}$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{pmatrix}, \quad \boldsymbol{\Lambda} = \begin{pmatrix} \boldsymbol{\Lambda}_{aa} & \boldsymbol{\Lambda}_{ab} \\ \boldsymbol{\Lambda}_{ba} & \boldsymbol{\Lambda}_{bb} \end{pmatrix},$$

then the conditional distribution $p(\mathbf{x}_a|\mathbf{x}_b)$ is given by

$$p(\mathbf{x}_a|\mathbf{x}_b) = \mathcal{N}(\mathbf{x}_a|\boldsymbol{\mu}_{a|b}, \boldsymbol{\Sigma}_{a|b}), \tag{B.5}$$

where

$$\begin{aligned} \boldsymbol{\mu}_{a|b} &= \boldsymbol{\mu}_a + \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}(\mathbf{x}_b - \boldsymbol{\mu}_b) \\ \boldsymbol{\Sigma}_{a|b} &= \boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba}, \end{aligned}$$

or alternatively, in terms of the precision matrix components

$$\begin{aligned} \boldsymbol{\mu}_{a|b} &= \boldsymbol{\mu}_a - \boldsymbol{\Lambda}_{aa}^{-1}\boldsymbol{\Lambda}_{ab}(\mathbf{x}_b - \boldsymbol{\mu}_b) \\ \boldsymbol{\Sigma}_{a|b} &= \boldsymbol{\Lambda}_{aa}^{-1}. \end{aligned}$$

B.3 Bayes rule

Given a marginal Gaussian distribution for $\mathbf{x}$ and a conditional Gaussian distribution for $\mathbf{y}$ given $\mathbf{x}$ in the form

$$\begin{aligned} p(\mathbf{x}) &= \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}) \\ p(\mathbf{y}|\mathbf{x}) &= \mathcal{N}(\mathbf{y}|\mathbf{A}\mathbf{x} + \mathbf{b}, \mathbf{L}^{-1}), \end{aligned}$$

the marginal distribution of $\mathbf{y}$ and the conditional distribution of $\mathbf{x}$ given $\mathbf{y}$ are given by

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^T + \mathbf{L}^{-1}) \tag{B.6}$$

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}|\mathbf{S}\{\mathbf{A}^T\mathbf{L}(\mathbf{y} - \mathbf{b}) + \boldsymbol{\Lambda}\boldsymbol{\mu}\}, \mathbf{S}), \tag{B.7}$$

where

$$\mathbf{S} = (\boldsymbol{\Lambda} + \mathbf{A}^T\mathbf{L}\mathbf{A})^{-1}.$$

B.4 Derivatives

Given a (multivariate) Gaussian distributions $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$, the derivative w.r.t. the mean $\boldsymbol{\mu}$ is given by

$$\frac{d\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})}{d\boldsymbol{\mu}} = -(\mathbf{x} - \boldsymbol{\mu})\boldsymbol{\Sigma}^{-1}\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}). \tag{B.8}$$

Similarly, the derivative w.r.t. the variance $\boldsymbol{\Sigma}$ is given by

$$\frac{d\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})}{d\boldsymbol{\Sigma}} = \left[\frac{1}{2\boldsymbol{\Sigma}^2}(\mathbf{x} - \boldsymbol{\mu})^T(\mathbf{x} - \boldsymbol{\mu}) - \frac{D}{2\boldsymbol{\Sigma}}\right]\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}). \tag{B.9}$$

Appendix C

Detailed derivations

C.1 Moments derivation for GPC-MA

Recall that the product of the cavity distribution with the exact likelihood term is given by

$$\begin{aligned}\hat{q}(f_n) &\triangleq \hat{Z}_n \mathcal{N}(\hat{\mu}_n, \hat{\sigma}_n^2) \\ &\simeq q_{-n}(f_n) \sum_{c_n \in \{0,1\}} p(\mathbf{y}_n|c_n)\, p(c_n|f_n),\end{aligned}$$

which, by making use of the definitions of the different probabilities, can be manipulated to give

$$\hat{q}(f_n) = b_n \mathcal{N}(f_n|\mu_{-n}, \sigma_{-n}^2) + (a_n - b_n)\, \Phi(f_n)\, \mathcal{N}(f_n|\mu_{-n}, \sigma_{-n}^2), \tag{C.1}$$

whose moments we wish to compute for moment matching.

In order to make the notation simpler and the derivation easier to follow, we will derive the moments using a "generic" distribution $q(x)$

$$q(x) = \frac{1}{Z}\Big[b\mathcal{N}(x|\mu, \sigma^2) + (a+b)\, \Phi(x)\, \mathcal{N}(x|\mu, \sigma^2)\Big]. \tag{C.2}$$

The normalization constant Z is given by

$$\begin{aligned}Z &= \int_{-\infty}^{+\infty} b\mathcal{N}(x|\mu, \sigma^2) + (a-b)\, \Phi(x)\, \mathcal{N}(x|\mu, \sigma^2)\, dx \\ &= b + (a-b) \underbrace{\int_{-\infty}^{+\infty} \Phi(x)\, \mathcal{N}(x|\mu, \sigma^2)\, dx}_{=\Phi(\eta)} \\ &= b + (a-b)\, \Phi(\eta),\end{aligned} \tag{C.3}$$

where

$$\eta = \frac{\mu}{\sqrt{1+\sigma^2}}.$$

Differentiating both sides with respect to μ gives

$$\begin{aligned}
&\frac{\partial Z}{\partial \mu} = \frac{\partial \Big[b + (a-b)\,\Phi(\eta) \Big]}{\partial \mu} \\
&\Leftrightarrow b \int \frac{x-\mu}{\sigma^2}\,\mathcal{N}(x|\mu,\sigma^2)\,dx + (a-b)\int \frac{x-\mu}{\sigma^2}\,\Phi(x)\,\mathcal{N}(x|\mu,\sigma^2)\,dx = \frac{(a-b)\,\mathcal{N}(\eta)}{\sqrt{1+\sigma^2}} \\
&\Leftrightarrow \frac{b}{\sigma^2}\int x\mathcal{N}(x|\mu,\sigma^2)\,dx - \frac{b\,\mu}{\sigma^2}\int \mathcal{N}(x|\mu,\sigma^2)\,dx \\
&\qquad + \frac{(a-b)}{\sigma^2}\int x\,\Phi(x)\,\mathcal{N}(x|\mu,\sigma^2)\,dx - \frac{(a-b)\,\mu}{\sigma^2}\int \Phi(x)\,\mathcal{N}(x|\mu,\sigma^2)\,dx = \frac{(a-b)\,\mathcal{N}(\eta)}{\sqrt{1+\sigma^2}} \\
&\Leftrightarrow \int x\Big[b\mathcal{N}(x|\mu,\sigma^2) + (a-b)\,\Phi(x)\,\mathcal{N}(x|\mu,\sigma^2) \Big]\,dx \\
&\qquad - \mu \underbrace{\int b\mathcal{N}(x|\mu,\sigma^2) + (a-b)\,\Phi(x)\,\mathcal{N}(x|\mu,\sigma^2)\,dx}_{=Z} = \frac{(a-b)\,\sigma^2\,\mathcal{N}(\eta)}{\sqrt{1+\sigma^2}},
\end{aligned}$$

where we made use of the fact that $\partial\Phi(\eta)/\partial\mu = \mathcal{N}(\eta)\,\partial\eta/\partial\mu$.

We recognise the first term on the left-hand side to be Z times the first moment of q, which we are seeking. Dividing through by Z gives

$$\mathbb{E}_q[x] = \mu + \frac{(a-b)\,\sigma^2\,\mathcal{N}(\eta)}{Z\,\sqrt{1+\sigma^2}} = \mu + \frac{(a-b)\,\sigma^2\,\mathcal{N}(\eta)}{\Big[b + (a-b)\,\Phi(\eta) \Big]\sqrt{1+\sigma^2}}. \tag{C.4}$$

Similarly, the second moment can be obtained by differentiating both sides of (C.3) twice to give

$$\begin{aligned}
&\frac{\partial^2 Z}{\partial^2 \mu} = \frac{\partial^2 \Big[b + (a-b)\,\Phi(\eta) \Big]}{\partial^2 \mu} \\
&\Leftrightarrow b \int \Big[\frac{x^2}{\sigma^4} - \frac{2\mu x}{\sigma^4} + \frac{\mu^2}{\sigma^4} - \frac{1}{\sigma^2} \Big]\,\mathcal{N}(x|\mu,\sigma^2)\,dx \\
&\qquad + (a-b)\int \Big[\frac{x^2}{\sigma^4} - \frac{2\mu x}{\sigma^4} + \frac{\mu^2}{\sigma^4} - \frac{1}{\sigma^2} \Big]\,\Phi(x)\,\mathcal{N}(x|\mu,\sigma^2)\,dx = -\frac{(a-b)\,\eta\,\mathcal{N}(\eta)}{1+\sigma^2}.
\end{aligned}$$

Multiplying through σ^4 and re-arranging gives

$$\begin{aligned}
&\Leftrightarrow b \int \Big[x^2 - 2\mu x + \mu^2 - \sigma^2 \Big]\,\mathcal{N}(x|\mu,\sigma^2)\,dx \\
&\qquad + (a-b)\int \Big[x^2 - 2\mu x + \mu^2 - \sigma^2 \Big]\,\Phi(x)\,\mathcal{N}(x|\mu,\sigma^2)\,dx = -\frac{(a-b)\,\sigma^4\,\eta\,\mathcal{N}(\eta)}{1+\sigma^2} \\
&\Leftrightarrow b \int x^2\mathcal{N}(x|\mu,\sigma^2)\,dx - 2\mu b\int x\,\mathcal{N}(x|\mu,\sigma^2)\,dx \\
&\qquad + \mu^2 b\int \mathcal{N}(x|\mu,\sigma^2)\,dx - \sigma^2 b\int \mathcal{N}(x|\mu,\sigma^2)\,dx \\
&\qquad + (a-b)\int x^2\Phi(x)\,\mathcal{N}(x|\mu,\sigma^2)\,dx - 2\mu(a-b)\int x\,\Phi(x)\,\mathcal{N}(x|\mu,\sigma^2)\,dx \\
&\qquad + \mu^2(a-b)\int \Phi(x)\,\mathcal{N}(x|\mu,\sigma^2)\,dx \\
&\qquad - \sigma^2(a-b)\int \Phi(x)\,\mathcal{N}(x|\mu,\sigma^2)\,dx = -\frac{(a-b)\,\sigma^4\,\eta\,\mathcal{N}(\eta)}{1+\sigma^2}
\end{aligned}$$

$$
\begin{aligned}
\Leftrightarrow & \int x^2 \Big[b\,\mathcal{N}(x|\mu,\sigma^2) + (a-b)\,\Phi(x)\,\mathcal{N}(x|\mu,\sigma^2) \Big]\, dx \\
& - 2\mu \int x \Big[b\,\mathcal{N}(x|\mu,\sigma^2) + (a-b)\,\Phi(x)\,\mathcal{N}(x|\mu,\sigma^2) \Big]\, dx \\
& + \mu^2 \underbrace{\int b\,\mathcal{N}(x|\mu,\sigma^2) + (a-b)\,\Phi(x)\,\mathcal{N}(x|\mu,\sigma^2)\, dx}_{=Z} \\
& - \sigma^2 \underbrace{\int b\,\mathcal{N}(x|\mu,\sigma^2) + (a-b)\,\Phi(x)\,\mathcal{N}(x|\mu,\sigma^2)\, dx}_{=Z} = -\frac{(a-b)\,\sigma^4\,\eta\,\mathcal{N}(\eta)}{1+\sigma^2} \\
\Leftrightarrow\ & Z\,\mathbb{E}_q[x^2] - 2\mu\,Z\,\mathbb{E}_q[x] + \mu^2\,Z - \sigma^2\,Z = -\frac{(a-b)\,\sigma^4\,\eta\,\mathcal{N}(\eta)}{1+\sigma^2}.
\end{aligned}
$$

Dividing through Z gives

$$
\begin{aligned}
& \Leftrightarrow \mathbb{E}_q[x^2] - 2\mu\,\mathbb{E}_q[x] + \mu^2 - \sigma^2 = -\frac{(a-b)\,\sigma^4\,\eta\,\mathcal{N}(\eta)}{Z\,(1+\sigma^2)} \\
& \Leftrightarrow \mathbb{E}_q[x^2] = 2\mu\,\mathbb{E}_q[x] - \mu^2 + \sigma^2 - \frac{(a-b)\,\sigma^4\,\eta\,\mathcal{N}(\eta)}{Z\,(1+\sigma^2)}. \qquad \text{(C.5)}
\end{aligned}
$$

The second moment is then given by

$$
\begin{aligned}
\mathbb{E}_q[(x-\mathbb{E}_q[x])^2] &= \mathbb{E}_q[x^2] - \mathbb{E}_q[x]^2 \\
&= 2\mu\,\mathbb{E}_q[x] - \mu^2 + \sigma^2 - \frac{(a-b)\,\sigma^4\,\eta\,\mathcal{N}(\eta)}{Z\,(1+\sigma^2)} - \left(\mu + \frac{(a-b)\,\sigma^2\,\mathcal{N}(\eta)}{Z\,\sqrt{1+\sigma^2}}\right)^2 \\
&= 2\mu\left(\mu + \frac{(a-b)\,\sigma^2\,\mathcal{N}(\eta)}{Z\,\sqrt{1+\sigma^2}}\right) \\
&\quad - \mu^2 + \sigma^2 - \frac{(a-b)\,\sigma^4\,\eta\,\mathcal{N}(\eta)}{Z\,(1+\sigma^2)} - \left(\mu + \frac{(a-b)\,\sigma^2\,\mathcal{N}(\eta)}{Z\,\sqrt{1+\sigma^2}}\right)^2 \\
&= 2\mu^2 + \frac{2\mu\,(a-b)\,\sigma^2\,\mathcal{N}(\eta)}{Z\,\sqrt{1+\sigma^2}} \\
&\quad - \mu^2 + \sigma^2 - \frac{(a-b)\,\sigma^4\,\eta\,\mathcal{N}(\eta)}{Z\,(1+\sigma^2)} - \left(\mu + \frac{(a-b)\,\sigma^2\,\mathcal{N}(\eta)}{Z\,\sqrt{1+\sigma^2}}\right)^2 \\
&= \mu^2 + \frac{2\mu\,(a-b)\,\sigma^2\,\mathcal{N}(\eta)}{Z\,\sqrt{1+\sigma^2}} + \sigma^2 - \frac{(a-b)\,\sigma^4\,\eta\,\mathcal{N}(\eta)}{Z\,(1+\sigma^2)} \\
&\quad - \left(\mu + \frac{(a-b)\,\sigma^2\,\mathcal{N}(\eta)}{Z\,\sqrt{1+\sigma^2}}\right)^2.
\end{aligned}
$$

Manipulating this expression further, we arrive at

$$
\mathbb{E}_q[(x-\mathbb{E}_q[x])^2] = \sigma^2 - \frac{\sigma^4}{1+\sigma^2}\left(\frac{\eta\,\mathcal{N}(\eta)\,(a-b)}{b+(a-b)\,\Phi(\eta)} + \frac{\mathcal{N}(\eta)^2\,(a-b)^2}{(b+(a-b)\,\Phi(\eta))^2}\right). \qquad \text{(C.6)}
$$

By making use of the moments derived above, the moments of the distribution

in eq. C.1 are then given by

$$\begin{aligned}
\hat{Z}_n &= b_n + (a_n - b_n)\,\Phi(\eta_n) \\
\hat{\mu}_n &= \mu_{-n} + \frac{(a_n - b_n)\,\sigma_{-n}^2\,\mathcal{N}(\eta_i)}{\Big[b_n + (a_n - b_n)\,\Phi(\eta_n)\Big]\sqrt{1-\sigma_{-n}^2}} \\
\hat{\sigma}_n &= \sigma_{-n}^2 - \frac{\sigma_{-n}^4}{1+\sigma_{-n}^2}\left(\frac{\eta_n\,\mathcal{N}(\eta_i)\,(a_n - b_n)}{b_n + (a_n - b_n)\,\Phi(\eta_n)} + \frac{\mathcal{N}(\eta_n)^2\,(a_n - b_n)^2}{(b_n + (a_n - b_n)\,\Phi(\eta_n))^2}\right),
\end{aligned}$$

where

$$\eta_n = \frac{\mu_{-n}}{\sqrt{1+\sigma_{-n}^2}}.$$

C.2 Variational inference for MA-sLDAc

Deriving the lower bound

The variational objective function (or the evidence lower bound) is given by

$$\begin{aligned}
&\log p(\mathbf{W}, \mathbf{Y}|\alpha, \tau, \omega, \boldsymbol{\eta}) \\
&\quad= \log \sum_{\mathbf{z}} \sum_{c} \frac{p(\boldsymbol{\theta}_{1:D}, \mathbf{z}_{1:D}, \boldsymbol{c}, \boldsymbol{\beta}_{1:K}, \boldsymbol{\Pi}_{1:R}, \mathbf{W}, \mathbf{Y}|\Theta)\, q(\boldsymbol{\theta}_{1:D}, \mathbf{z}_{1:D}, \boldsymbol{c}, \boldsymbol{\beta}_{1:K}, \boldsymbol{\Pi}_{1:R})}{q(\boldsymbol{\theta}_{1:D}, \mathbf{z}_{1:D}, \boldsymbol{c}, \boldsymbol{\beta}_{1:K}, \boldsymbol{\Pi}_{1:R})} \\
&\quad\geqslant \mathcal{L}(\mathbf{W}, \mathbf{Y}|\Theta) \\
&\quad= \mathbb{E}_q[\log p(\boldsymbol{\theta}_{1:D}, \mathbf{z}_{1:D}, \boldsymbol{c}, \boldsymbol{\beta}_{1:K}, \boldsymbol{\Pi}_{1:R}, \mathbf{W}, \mathbf{Y}|\Theta)] \underbrace{-\mathbb{E}_q[\log q(\boldsymbol{\theta}_{1:D}, \mathbf{z}_{1:D}, \boldsymbol{c}, \boldsymbol{\beta}_{1:K}, \boldsymbol{\Pi}_{1:R})]}_{\mathcal{H}(q)} \\
&\quad= \sum_{i=1}^{K} \mathbb{E}_q[\log p(\boldsymbol{\beta}_i|\tau\mathbf{1}_V)] + \sum_{r=1}^{R}\sum_{c=1}^{C} \mathbb{E}_q[\log p(\boldsymbol{\pi}_c^r|\omega\mathbf{1}_C)] \\
&\quad+ \sum_{d=1}^{D}\Bigg(\mathbb{E}_q[\log p(\boldsymbol{\theta}^d|\alpha\mathbf{1}_K)] + \sum_{n=1}^{N^d} \mathbb{E}_q[\log p(z_n^d|\boldsymbol{\theta}^d)] + \sum_{n=1}^{N^d} \mathbb{E}_q[\log p(w_n^d|z_n^d, \boldsymbol{\beta}_{1:K})] \\
&\quad+ \mathbb{E}_q[\log p(c^d|\bar{\mathbf{z}}^d, \boldsymbol{\eta})] + \sum_{r=1}^{R} \mathbb{E}_q[\log p(y^{d,r}|c^d, \boldsymbol{\Pi}^r)]\Bigg) + \mathcal{H}(q)
\end{aligned} \tag{C.7}$$

where the entropy $\mathcal{H}(q)$ of the variational distribution is given by

$$\begin{aligned}
\mathcal{H}(q) = &- \sum_{r=1}^{R}\sum_{c=1}^{C} \mathbb{E}_q[\log q(\boldsymbol{\pi}_c^r|\boldsymbol{\xi}_c^r)] - \sum_{i=1}^{K} \mathbb{E}_q[\log q(\boldsymbol{\beta}_i|\boldsymbol{\zeta}_i)] \\
&- \sum_{d=1}^{D}\Bigg(\mathbb{E}_q[\log q(\boldsymbol{\theta}^d|\boldsymbol{\gamma}^d)] - \sum_{n=1}^{N^d} \mathbb{E}_q[\log q(z_n^d|\boldsymbol{\phi}_n^d)] - \mathbb{E}_q[\log q(c^d|\boldsymbol{\lambda}^d)]\Bigg).
\end{aligned} \tag{C.8}$$

The terms needed for the lower bound are given by

$$\mathbb{E}_q[\log p(\boldsymbol{\beta}_i|\tau\mathbf{1}_V)] = \mathbb{E}_q\left[\log \frac{\Gamma(\tau V)}{\prod_{j=1}^V \Gamma(\tau)} \prod_{j=1}^V \beta_{i,j}^{(\tau-1)}\right]$$
$$= \log\Gamma(\tau V) - \sum_{j=1}^V \log\Gamma(\tau) + \sum_{j=1}^V (\tau-1)\,\mathbb{E}_q[\log\beta_{i,j}]$$

$$\mathbb{E}_q[\log p(\boldsymbol{\pi}_c^r|\omega\mathbf{1}_C)] = \mathbb{E}_q\left[\log \frac{\Gamma(\omega C)}{\prod_{l=1}^C \Gamma(\omega)} \prod_{l=1}^C \left(\pi_{c,l}^r\right)^{(\omega-1)}\right]$$
$$= \log\Gamma(\omega C) - \sum_{l=1}^C \log\Gamma(\omega) + \sum_{l=1}^C (\omega-1)\,\mathbb{E}_q[\log\pi_{c,l}^r]$$

$$\mathbb{E}_q[\log p(\boldsymbol{\theta}^d|\alpha\mathbf{1}_K)] = \mathbb{E}_q\left[\log \frac{\Gamma(\alpha K)}{\prod_{i=1}^K \Gamma(\alpha)} \prod_{i=1}^K (\theta_i^d)^{(\alpha-1)}\right]$$
$$= \log\Gamma(\alpha K) - \sum_{i=1}^K \log\Gamma(\alpha) + \sum_{i=1}^K (\alpha-1)\,\mathbb{E}_q[\log\theta_i^d]$$

$$\mathbb{E}_q[\log p(z_n^d|\boldsymbol{\theta}^d)] = \mathbb{E}_q\left[\log \prod_{i=1}^K \left(\theta_i^d\right)^{z_{n,i}^d}\right] = \sum_{i=1}^K \phi_{n,i}^d\,\mathbb{E}_q[\log\theta_i^d]$$

$$\mathbb{E}_q[\log p(w_n^d|z_n^d,\boldsymbol{\beta}_{1:K})] = \mathbb{E}_q\left[\log \prod_{j=1}^V (\beta_{z_n^d,j})^{w_{n,j}^d}\right] = \sum_{j=1}^V \sum_{i=1}^K w_{n,j}^d \phi_{n,i}^d\,\mathbb{E}_q[\log\beta_{i,j}]$$

$$\mathbb{E}_q[\log p(y^{d,r}|c^d,\boldsymbol{\Pi}^r)] = \mathbb{E}_q\left[\log \prod_{l=1}^C (\pi_{c^d,l}^r)^{y_l^{d,r}}\right] = \sum_{c=1}^C \sum_{l=1}^C \lambda_c^d y_l^{d,r}\,\mathbb{E}_q[\log\pi_{c,l}^r]$$

$$\mathbb{E}_q[\log q(\boldsymbol{\pi}_c^r|\boldsymbol{\xi}_c^r)] = \mathbb{E}_q\left[\log \frac{\Gamma(\sum_{t=1}^C \xi_{c,t}^r)}{\prod_{l=1}^C \Gamma(\xi_{c,l}^r)} \prod_{l=1}^C \left(\pi_{c,l}^r\right)^{(\xi_{c,l}^r-1)}\right]$$
$$= \log\Gamma\left(\sum_{t=1}^C \xi_{c,t}^r\right) - \sum_{l=1}^C \log\Gamma(\xi_{c,l}^r) + \sum_{l=1}^C (\xi_{c,l}^r-1)\,\mathbb{E}_q[\log\pi_{c,l}^r]$$

$$\mathbb{E}_q[\log q(\boldsymbol{\beta}_i|\boldsymbol{\zeta}_i)] = \mathbb{E}_q\left[\log \frac{\Gamma(\sum_{k=1}^V \zeta_{i,k})}{\prod_{j=1}^V \Gamma(\zeta_{i,j})} \prod_{j=1}^V \left(\beta_{i,j}\right)^{(\zeta_{i,j}-1)}\right]$$
$$= \log\Gamma\left(\sum_{k=1}^V \zeta_{i,k}\right) - \sum_{j=1}^V \log\Gamma(\zeta_{i,j}) + \sum_{j=1}^V (\zeta_{i,j}-1)\,\mathbb{E}_q[\log\beta_{i,j}]$$

$$\mathbb{E}_q[\log q(\boldsymbol{\theta}^d|\boldsymbol{\gamma}^d)] = \mathbb{E}_q\left[\log \frac{\Gamma(\sum_{j=1}^K \gamma_j^d)}{\prod_{i=1}^K \Gamma(\gamma_i^d)} \prod_{i=1}^K \left(\theta_i^d\right)^{(\gamma_i^d-1)}\right]$$
$$= \log\Gamma\left(\sum_{j=1}^K \gamma_j^d\right) - \sum_{i=1}^K \log\Gamma(\gamma_i^d) + \sum_{i=1}^K (\gamma_i^d-1)\,\mathbb{E}_q[\log\theta_i^d]$$

$$\mathbb{E}_q[\log q(z_n^d|\boldsymbol{\phi}_n^d)] = \mathbb{E}_q\left[\log \prod_{i=1}^K \left(\phi_{n,i}^d\right)^{z_{n,i}^d}\right] = \sum_{i=1}^K \phi_{n,i}^d \log\phi_{n,i}^d$$

$$\mathbb{E}_q[\log q(c^d|\boldsymbol{\lambda}^d)] = \mathbb{E}_q\left[\log \prod_{l=1}^C \left(\lambda_l^d\right)^{c_l^d}\right] = \sum_{l=1}^C \lambda_l^d \log\lambda_l^d,$$

where

$$\mathbb{E}_q[\log \theta_i^d] = \Psi(\gamma_i^d) - \Psi\Big(\sum_{j=1}^{K} \gamma_j^d\Big)$$
$$\mathbb{E}_q[\log \beta_{i,j}] = \Psi(\zeta_{i,j}) - \Psi\Big(\sum_{k=1}^{V} \zeta_{i,k}\Big)$$
$$\mathbb{E}_q[\log \pi_{c,l}^r] = \Psi(\xi_{c,l}^r) - \Psi\Big(\sum_{t=1}^{C} \xi_{c,t}^r\Big).$$

Finally, the expectation of the log probability of the latent classes is given by

$$\begin{aligned}\mathbb{E}_q[\log p(c^d|\bar{\mathbf{z}}^d, \boldsymbol{\eta})] &= \mathbb{E}_q\Big[\log \frac{\exp(\boldsymbol{\eta}_{c^d}^T \bar{\mathbf{z}}^d)}{\sum_{l=1}^{C} \exp(\boldsymbol{\eta}_l^T \bar{\mathbf{z}}^d)}\Big] \\ &= \mathbb{E}_q[\boldsymbol{\eta}_{c^d}^T \bar{\mathbf{z}}^d] - \mathbb{E}_q\Big[\log \sum_{l=1}^{C} \exp(\boldsymbol{\eta}_l^T \bar{\mathbf{z}}^d)\Big],\end{aligned}$$

where the first term can be easily computed as $\mathbb{E}_q[\boldsymbol{\eta}_{c^d}^T \bar{\mathbf{z}}^d] = \sum_{l=1}^{C} \lambda_l^d \boldsymbol{\eta}_l^T \bar{\boldsymbol{\phi}}^d$ and the second term can be lower-bounded by appealing again to the Jensen's inequality as follows

$$\begin{aligned}-\mathbb{E}_q\Big[\log \sum_{l=1}^{C} \exp(\boldsymbol{\eta}_l^T \bar{\mathbf{z}}^d)\Big] &\geqslant -\log \sum_{l=1}^{C} \mathbb{E}_q[\exp(\boldsymbol{\eta}_l^T \bar{\mathbf{z}}^d)] \\ &= -\log \sum_{l=1}^{C} \mathbb{E}_q\Big[\exp(\boldsymbol{\eta}_l^T \frac{1}{N^d} \sum_{j=1}^{N^d} z_j^d)\Big] \\ &= -\log \sum_{l=1}^{C} \prod_{j=1}^{N^d} \big(\boldsymbol{\phi}_j^d\big)^T \exp\Big(\frac{1}{N^d}\boldsymbol{\eta}_l\Big) \\ &= -\log \sum_{l=1}^{C} \big(\boldsymbol{\phi}_n^d\big)^T \exp\Big(\frac{1}{N^d}\boldsymbol{\eta}_l\Big) \prod_{j=1, j\neq n}^{N^d} \big(\boldsymbol{\phi}_j^d\big)^T \exp\Big(\frac{1}{N^d}\boldsymbol{\eta}_l\Big) \\ &= -\log \big(\boldsymbol{\phi}_n^d\big)^T \underbrace{\sum_{l=1}^{C} \exp\Big(\frac{1}{N^d}\boldsymbol{\eta}_l\Big) \prod_{j=1, j\neq n}^{N^d} \big(\boldsymbol{\phi}_j^d\big)^T \exp\Big(\frac{1}{N^d}\boldsymbol{\eta}_l\Big)}_{=\mathbf{a}} \\ &= -\log \mathbf{a}^T \boldsymbol{\phi}_n^d, \end{aligned} \tag{C.9}$$

where we defined

$$\mathbf{a} = \sum_{l=1}^{C} \exp(\frac{1}{N^d}\boldsymbol{\eta}_l) \prod_{j=1, j\neq n}^{N^d} \big(\boldsymbol{\phi}_j^d\big)^T \exp\Big(\frac{1}{N^d}\boldsymbol{\eta}_l\Big).$$

Putting all the terms together, the lower-bound becomes

$$
\begin{aligned}
&\mathcal{L}(\mathbf{w}_{1:D}, \mathbf{y}_{1:D}|\Theta) \\
&= \sum_{i=1}^{K}\left(\log\Gamma(\tau V) - \sum_{j=1}^{V}\log\Gamma(\tau) + \sum_{j=1}^{V}(\tau-1)\left(\Psi(\zeta_{i,j}) - \Psi\Big(\sum_{k=1}^{V}\zeta_{i,k}\Big)\right)\right) \\
&+ \sum_{r=1}^{R}\sum_{c=1}^{C}\left(\log\Gamma(\omega C) - \sum_{l=1}^{C}\log\Gamma(\omega) + \sum_{l=1}^{C}(\omega-1)\left(\Psi(\xi_{c,l}^{r}) - \Psi\Big(\sum_{t=1}^{C}\xi_{c,t}^{r}\Big)\right)\right) \\
&+ \sum_{d=1}^{D}\left(\log\Gamma(\alpha K) - \sum_{i=1}^{K}\log\Gamma(\alpha) + \sum_{i=1}^{K}(\alpha-1)\left(\Psi(\gamma_i^d) - \Psi\Big(\sum_{j=1}^{K}\gamma_j^d\Big)\right)\right) \\
&+ \sum_{d=1}^{D}\sum_{n=1}^{N^d}\sum_{i=1}^{K}\phi_{n,i}^{d}\left(\Psi(\gamma_i^d) - \Psi\Big(\sum_{j=1}^{K}\gamma_j^d\Big)\right) \\
&+ \sum_{d=1}^{D}\sum_{n=1}^{N^d}\sum_{j=1}^{V}\sum_{i=1}^{K} w_{n,j}^{d}\phi_{n,i}^{d}\left(\Psi(\zeta_{i,j}) - \Psi\Big(\sum_{k=1}^{V}\zeta_{i,k}\Big)\right) \\
&+ \sum_{d=1}^{D}\left(\sum_{l=1}^{C}\lambda_l^d \boldsymbol{\eta}_l^T\bar{\boldsymbol{\phi}}^d - (\mathbf{a}^T(\boldsymbol{\phi}_n^d)^{old})^{-1}(\mathbf{a}^T\boldsymbol{\phi}_n^d) - \log(\mathbf{a}^T(\boldsymbol{\phi}_n^d)^{old}) + 1\right) \\
&+ \sum_{d=1}^{D}\sum_{r=1}^{R}\sum_{c=1}^{C}\sum_{l=1}^{C}\lambda_c^d y_l^{d,r}\left(\Psi(\xi_{c,l}^{r}) - \Psi\Big(\sum_{t=1}^{C}\xi_{c,t}^{r}\Big)\right) \\
&- \sum_{r=1}^{R}\sum_{c=1}^{C}\left(\log\Gamma\Big(\sum_{t=1}^{C}\xi_{c,t}^{r}\Big) - \sum_{l=1}^{C}\log\Gamma(\xi_{c,l}^{r}) + \sum_{l=1}^{C}(\xi_{c,l}^{r}-1)\left(\Psi(\xi_{c,l}^{r}) - \Psi\Big(\sum_{t=1}^{C}\xi_{c,t}^{r}\Big)\right)\right) \\
&- \sum_{i=1}^{K}\left(\log\Gamma\Big(\sum_{k=1}^{V}\zeta_{i,k}\Big) - \sum_{j=1}^{V}\log\Gamma(\zeta_{i,j}) + \sum_{j=1}^{V}(\zeta_{i,j}-1)\left(\Psi(\zeta_{i,j}) - \Psi\Big(\sum_{k=1}^{V}\zeta_{i,k}\Big)\right)\right) \\
&- \sum_{d=1}^{D}\left(\log\Gamma\Big(\sum_{j=1}^{K}\gamma_j^d\Big) - \sum_{i=1}^{K}\log\Gamma(\gamma_i^d) + \sum_{i=1}^{K}(\gamma_i^d-1)\left(\Psi(\gamma_i^d) - \Psi\Big(\sum_{j=1}^{K}\gamma_j^d\Big)\right)\right) \\
&- \sum_{d=1}^{D}\sum_{n=1}^{N^d}\sum_{i=1}^{K}\phi_{n,i}^{d}\log\phi_{n,i}^{d} \\
&- \sum_{d=1}^{D}\sum_{l=1}^{C}\lambda_l^d\log\lambda_l^d.
\end{aligned}
\tag{C.10}
$$

Optimizing the lower bound (E-step)

Optimizing w.r.t. γ_i^d

Collecting only the terms in the bound that contain $\boldsymbol{\gamma}$ gives

$$\mathcal{L}_{[\boldsymbol{\gamma}]} = \sum_{d=1}^{D}\sum_{i=1}^{K}\Psi(\gamma_i^d)\left(\alpha + \sum_{n=1}^{N^d}\phi_{n,i}^d - \gamma_i^d\right) - \sum_{d=1}^{D}\sum_{i=1}^{K}\Psi\Big(\sum_{j=1}^{K}\gamma_j^d\Big)\left(\alpha + \sum_{n=1}^{N^d}\phi_{n,i}^d - \gamma_i^d\right) - \sum_{d=1}^{D}\log\Gamma\Big(\sum_{j=1}^{K}\gamma_j^d\Big) + \sum_{d=1}^{D}\sum_{i=1}^{K}\log\Gamma(\gamma_i^d).$$

Taking derivatives w.r.t. γ_i^d gives

$$\frac{\partial\mathcal{L}_{[\boldsymbol{\gamma}]}}{\partial\gamma_i^d} = \Psi'(\gamma_i^d)\Big(\alpha + \sum_{n=1}^{N^d}\phi_{n,i}^d - \gamma_i^d\Big) - \Psi'\Big(\sum_{j=1}^{K}\gamma_j^d\Big)\sum_{j=1}^{K}\Big(\alpha + \sum_{n=1}^{N^d}\phi_{n,j}^d - \gamma_j^d\Big).$$

Setting this derivative to zero in order to get a maximum, we get the solution

$$\gamma_i^d = \alpha + \sum_{n=1}^{N^d}\phi_{n,i}^d. \tag{C.11}$$

Optimizing w.r.t. $\phi_{n,i}^d$

Collecting only the terms in the bound that contain $\boldsymbol{\phi}$ and adding Lagrange multipliers gives

$$\begin{aligned}\mathcal{L}_{[\boldsymbol{\phi}]} = &\sum_{d=1}^{D}\sum_{n=1}^{N^d}\sum_{i=1}^{K}\phi_{n,i}^d\Big(\Psi(\gamma_i^d) - \Psi\Big(\sum_{j=1}^{K}\gamma_j^d\Big)\Big)\\ &+ \sum_{d=1}^{D}\sum_{n=1}^{N^d}\sum_{j=1}^{V}\sum_{i=1}^{K}w_{n,j}^d\phi_{n,i}^d\left(\Psi(\zeta_{i,j}) - \Psi\Big(\sum_{k=1}^{V}\zeta_{i,k}\Big)\right)\\ &+ \sum_{d=1}^{D}\left(\frac{1}{N^d}\sum_{l=1}^{C}\lambda_l^d\sum_{n=1}^{N^d}\boldsymbol{\eta}_l^T\boldsymbol{\phi}_n^d - (\mathbf{a}^T(\boldsymbol{\phi}_n^d)^{old})^{-1}(\mathbf{a}^T\boldsymbol{\phi}_n^d) - \log(\mathbf{a}^T(\boldsymbol{\phi}_n^d)^{old})\right)\\ &- \sum_{d=1}^{D}\sum_{n=1}^{N^d}\sum_{i=1}^{K}\phi_{n,i}^d\log\phi_{n,i}^d + \mu\Big(\sum_{k=1}^{K}\phi_{n,k}^d - 1\Big).\end{aligned}$$

Taking derivatives w.r.t. $\phi_{n,i}^d$ gives

$$\begin{aligned}\frac{\partial\mathcal{L}_{[\boldsymbol{\phi}]}}{\partial\phi_{n,i}^d} = &\Psi(\gamma_i^d) - \Psi\Big(\sum_{j=1}^{K}\gamma_j^d\Big) + \sum_{j=1}^{V}w_{n,j}^d\Psi(\zeta_{i,j}) - \sum_{j=1}^{V}w_{n,j}^d\Psi\Big(\sum_{k=1}^{V}\zeta_{i,k}\Big)\\ &+ \frac{1}{N^d}\sum_{l=1}^{C}\lambda_l^d\eta_{l,i} - (\mathbf{a}^T(\boldsymbol{\phi}_n^d)^{old})^{-1}a_i - \log\phi_{n,i}^d - 1 + \mu.\end{aligned}$$

The updates for $\phi_{n,i}^d$ are then given by

$$\phi_{n,i}^d \propto \exp\left(\Psi(\gamma_i) + \sum_{j=1}^{V} w_{n,j}^d \Psi(\zeta_{i,j}) - \sum_{j=1}^{V} w_{n,j}^d \Psi\left(\sum_{k=1}^{V} \zeta_{i,k}\right) + \frac{1}{N^d} \sum_{l=1}^{C} \lambda_l^d \boldsymbol{\eta}_{l,i} - (\mathbf{a}^T (\boldsymbol{\phi}_n^d)^{old})^{-1} a_i\right). \tag{C.12}$$

Optimizing w.r.t. λ_l^d

Collecting only the terms in the bound that contain $\boldsymbol{\lambda}$ and adding Lagrange multipliers gives

$$\mathcal{L}_{[\boldsymbol{\lambda}]} = \sum_{d=1}^{D}\sum_{l=1}^{C} \lambda_l^d \boldsymbol{\eta}_l^T \bar{\boldsymbol{\phi}}^d + \sum_{d=1}^{D}\sum_{r=1}^{R}\sum_{l=1}^{C}\sum_{c=1}^{C} \lambda_l^d y_c^{d,r} \left(\Psi(\xi_{l,c}^r) - \Psi\left(\sum_{t=1}^{C} \xi_{l,t}^r\right)\right) - \sum_{l=1}^{C} \lambda_l^d \log \lambda_l^d + \mu\left(\sum_{k=1}^{C} \lambda_k^d - 1\right).$$

Taking derivatives w.r.t. λ_l^d gives

$$\frac{\partial \mathcal{L}_{[\boldsymbol{\lambda}]}}{\partial \lambda_l^d} = \boldsymbol{\eta}_l^T \bar{\boldsymbol{\phi}}^d + \sum_{r=1}^{R}\sum_{c=1}^{C} y_c^{d,r} \Psi(\xi_{l,c}^r) - \sum_{r=1}^{R}\sum_{c=1}^{C} y_c^{d,r} \Psi\left(\sum_{t=1}^{C} \xi_{l,t}^r\right) - \log \lambda_l^d - 1 + \mu.$$

The updates for λ_l^d are then given by

$$\lambda_l^d \propto \exp\left(\boldsymbol{\eta}_l^T \bar{\boldsymbol{\phi}}^d + \sum_{r=1}^{R}\sum_{c=1}^{C} y_c^{d,r} \Psi(\xi_{l,c}^r) - \sum_{r=1}^{R}\sum_{c=1}^{C} y_c^{d,r} \Psi\left(\sum_{t=1}^{C} \xi_{l,t}^r\right)\right). \tag{C.13}$$

Optimizing w.r.t. $\zeta_{i,j}$

Collecting only the terms in the bound that contain $\boldsymbol{\zeta}$ gives

$$\mathcal{L}_{[\boldsymbol{\zeta}]} = \sum_{i=1}^{K}\sum_{j=1}^{V} \left(\Psi(\zeta_{i,j}) - \Psi\left(\sum_{k=1}^{V} \zeta_{i,k}\right)\right)\left(\tau + \sum_{d=1}^{D}\sum_{n=1}^{N^d} w_{n,j}^d \phi_{n,i}^d - \zeta_{i,j}\right) - \sum_{i=1}^{K} \log \Gamma\left(\sum_{k=1}^{V} \zeta_{i,k}\right) + \sum_{i=1}^{K}\sum_{j=1}^{V} \log \Gamma(\zeta_{i,j}).$$

Taking derivatives w.r.t. $\zeta_{i,j}$ gives

$$\frac{\partial \mathcal{L}_{[\boldsymbol{\zeta}]}}{\partial \zeta_{i,j}} = \left(\Psi'(\zeta_{i,j}) - \Psi'\left(\sum_{k=1}^{V} \zeta_{i,k}\right)\right)\left(\tau + \sum_{d=1}^{D}\sum_{n=1}^{N^d} w_{n,j}^d \phi_{n,i}^d - \zeta_{i,j}\right).$$

Setting this derivative to zero in order to get a maximum, we get the solution

$$\zeta_{i,j} = \tau + \sum_{d=1}^{D}\sum_{n=1}^{N^d} w_{n,j}^d \phi_{n,i}^d. \tag{C.14}$$

Optimizing w.r.t. $\xi_{c,l}^r$

Collecting only the terms in the bound that contain $\boldsymbol{\xi}$ gives

$$\mathcal{L}_{[\boldsymbol{\xi}]} = \sum_{r=1}^{R}\sum_{c=1}^{C}\sum_{l=1}^{C}\left(\Psi(\xi_{c,l}^r) - \Psi\Big(\sum_{t=1}^{C}\xi_{c,t}^r\Big)\right)\left(\omega + \sum_{d=1}^{D}\lambda_c^d y_l^{d,r} - \xi_{c,l}^r\right) - \sum_{r=1}^{R}\sum_{c=1}^{C}\log\Gamma\Big(\sum_{t=1}^{C}\xi_{c,t}^r\Big) + \sum_{r=1}^{R}\sum_{c=1}^{C}\sum_{l=1}^{C}\log\Gamma(\xi_{c,l}^r).$$

Taking derivatives w.r.t. $\xi_{c,l}^r$ gives

$$\frac{\partial\mathcal{L}_{[\boldsymbol{\xi}]}}{\partial\xi_{c,l}^r} = \left(\Psi'(\xi_{c,l}^r) - \Psi'\Big(\sum_{t=1}^{C}\xi_{c,t}^r\Big)\right)\left(\omega + \sum_{d=1}^{D}\lambda_c^d y_l^{d,r} - \xi_{c,l}^r\right).$$

Setting this derivative to zero in order to get a maximum, we get the solution

$$\xi_{c,l}^r = \omega + \sum_{d=1}^{D}\lambda_c^d y_l^{d,r}. \tag{C.15}$$

Parameter estimation (M-step)

Given a corpus of D documents labeled by R different annotators, $\mathcal{D} = \{\mathbf{w}^d, \mathbf{y}^d\}_{d=1}^D$, we find maximum likelihood estimates for the class coefficients $\boldsymbol{\eta}$ by maximizing the lower bound on the log-likelihood w.r.t. $\boldsymbol{\eta}$. Collecting only the terms in the bound that contain $\boldsymbol{\eta}_l$ gives

$$\mathcal{L}_{[\boldsymbol{\eta}]} = \sum_{d=1}^{D}\left(\sum_{l=1}^{C}\lambda_l^d\boldsymbol{\eta}_l^T\bar{\boldsymbol{\phi}}^d - \log\sum_{l=1}^{C}\prod_{j=1}^{N^d}\Big(\sum_{i=1}^{K}\phi_{j,i}^d\exp\Big(\frac{1}{N^d}\eta_{l,i}\Big)\Big)\right).$$

Taking derivatives w.r.t. $\eta_{l,i}$ gives

$$\frac{\partial\mathcal{L}_{[\boldsymbol{\eta}]}}{\partial\eta_{l,i}} = \sum_{d=1}^{D}\left(\lambda_{l,i}^d\bar{\boldsymbol{\phi}}_i^d - \frac{\prod_{n=1}^{N^d}\Big(\sum_{i=1}^{K}\phi_{n,i}^d\exp\Big(\frac{1}{N^d}\eta_{l,i}\Big)\Big)}{\sum_{t=1}^{C}\prod_{n=1}^{N^d}\Big(\sum_{i=1}^{K}\phi_{n,i}^d\exp\Big(\frac{1}{N^d}\eta_{t,i}\Big)\Big)}\sum_{n=1}^{N^d}\frac{\frac{1}{N^d}\phi_{n,i}^d\exp(\frac{1}{N^d}\eta_{l,i})}{\sum_{j=1}^{K}\phi_{n,j}^d\exp(\frac{1}{N^d}\eta_{l,j})}\right).$$

Setting this derivative to zero does not lead to a closed-form solution, hence a numerical optimization routine (L-BFGS) is used.

C.3 Expectation propagation for BAM-GP

In this appendix, we exploit the message-passing viewpoint of EP, to present an algorithm for performing approximate inference in the proposed Bayesian additive model by passing messages in the factor graph of Fig. 7.2. We adopt the following notation for the messages: we denote the message sent from factor f to variable x in iteration t as $m_{f\to x}^t(x)$; similarly, message sent from variable x to factor f in iteration t is $m_{x\to f}^t(x)$. All messages correspond to the Gaussian distributions with mean μ and variance v such that, for example, the message $m_{x\to f}^t(x) = \mathcal{N}(x|\mu_{x\to f}^t, v_{x\to f}^t)$.

Let us start by assigning names to the factors in Figure 7.2:

$$\begin{aligned}
g^r(\mathbf{f}^r) &= \mathcal{GP}(0, k_r(\mathbf{x}^r, \mathbf{x}^{r'})) \\
g^e(\mathbf{f}^e) &= \mathcal{GP}(0, k_e(\mathbf{x}^e, \mathbf{x}^{e'})) \\
l_n^r(f_n^r) &= \mathbb{I}(y_n^r > 0) \\
l_n^{e_i}(f_n^{e_i}) &= \mathbb{I}(y_n^{e_i} > 0) \\
h_n^r(y_n^r, f_n^r) &= \mathcal{N}(y_n^r | f_n^r, \beta_r) \\
h_n^{e_i}(y_n^{e_i}, f_n^{e_i}) &= \mathcal{N}(y_n^{e_i} | f_n^{e_i}, \beta_e) \\
k_n(y_n, y_n^r, \{y_n^{e_i}\}_{i=1}^{E_n}) &= \mathcal{N}(y_n | y_n^r + \textstyle\sum_{i=1}^{E_n} y_n^{e_i}, v).
\end{aligned}$$

The steps of the EP algorithm are then the following:

Step 1: Compute message from the g^r and g^e factors to the f_n^r and f_n^e variables respectively

$$\begin{aligned}
m_{g^r \to f_n^r}^t(f_n^r) &= \int p(\mathbf{f}^r|\mathbf{X}) \prod_{j \neq n} m_{f_j^r \to g^r}^{t-1}(f_j^r)\, df_j^r. \\
&= \int p(\mathbf{f}^r|\mathbf{X}) \prod_{j \neq i} \mathcal{N}\Big(f_j^r \Big| \mu_{f_j^r \to g^r}^{t-1}, v_{f_j^r \to g^r}^{t-1}\Big)\, df_j^r
\end{aligned} \tag{C.16}$$

Conceptually, one can think of the combination of prior $p(\mathbf{f}^r|\mathbf{X}) = \mathcal{N}(\mathbf{f}^r|\mathbf{0}, \mathbf{K}^r)$ and the $n-1$ (approximate) messages in (C.16) in two ways, either by explicitly multiplying out the factors, or (equivalently) by removing the n^{th} message from the approximate posterior on $\mathbf{f}^r$. Here, we follow the latter approach. The approximate posterior on $\mathbf{f}^r$ is given by

$$\begin{aligned}
q(\mathbf{f}^r) &= \frac{1}{Z_{EP}} \mathcal{N}(\mathbf{f}^r|\mathbf{0}, \mathbf{K}^r) \prod_{n=1}^{N} \mathcal{N}\Big(f_n^r \Big| \mu_{f_n^r \to g^r}^{t-1}, v_{f_n^r \to g^r}^{t-1}\Big) \\
&= \mathcal{N}(\mathbf{f}^r|\boldsymbol{\mu}^r, \boldsymbol{\Sigma}^r),
\end{aligned}$$

with

$$\begin{aligned}
\boldsymbol{\mu}^r &= \boldsymbol{\Sigma}^r \big(\tilde{\boldsymbol{\Sigma}}^r\big)^{-1} \tilde{\boldsymbol{\mu}}^r, \\
\boldsymbol{\Sigma}^r &= \big(\big(\mathbf{K}^r\big)^{-1} + \big(\tilde{\boldsymbol{\Sigma}}^r\big)^{-1}\big)^{-1},
\end{aligned}$$

where $\tilde{\boldsymbol{\mu}}^r$ is the vector of $\mu_{f_n^r \to g^r}^{t-1}$ and $\tilde{\boldsymbol{\Sigma}}^r$ is a diagonal matrix with $\tilde{\Sigma}_{nn}^r = v_{f_n^r \to g^r}^{t-1}$. Hence, the marginal for f_n^r from $q^r(\mathbf{f}^r)$ is given by

$$q(f_n^r) = \mathcal{N}\big(f_n^r \big| \mu_n^r, \Sigma_{nn}^r\big).$$

The message from the factor $g^r(\mathbf{f}^r)$ to the f_n^r variables is then given by

$$\begin{aligned}
m_{g^r \to f_n^r}^t(f_n^r) &= \frac{q(f_n^r)}{m_{f_n^r \to g^r}^{t-1}(f_n^r)} = \frac{\mathcal{N}\Big(f_n^r \Big| \mu_n^r, \Sigma_{nn}^r\Big)}{\mathcal{N}\Big(f_n^r \Big| \mu_{f_n^r \to g^r}^{t-1}, v_{f_n^r \to g^r}^{t-1}\Big)} = \mathcal{N}\Big(f_n \Big| \mu_{g^r \to f_n^r}^t, v_{g^r \to f_n^r}^t\Big), \\
\mu_{g^r \to f_n^r}^t &= v_{g^r \to f_n^r}\Big(\mu_n^r / \Sigma_{nn}^r - \mu_{f_n^r \to g^r}^{t-1} / v_{f_n^r \to g^r}^{t-1}\Big), \\
v_{g^r \to f_n^r}^t &= \Big(1/\Sigma_{nn}^r - 1/v_{f_n^r \to g^r}^{t-1}\Big)^{-1}.
\end{aligned}$$

Step 2: Compute the posterior on f_n^r as the product of all incoming messages

$$\begin{aligned}
q^t(f_n^r) &= m^t_{g^r \to f_n^r}(f_n^r)\, m^{t-1}_{h_n^r \to f_n^r}(f_n^r) \\
&= \mathcal{N}\Big(f_n^r \Big| \mu^t_{g^r \to f_n^r}, v^t_{g^r \to f_n^r}\Big)\, \mathcal{N}\Big(f_n^r \Big| \mu^{t-1}_{h_n^r \to f_n^r}, v^{t-1}_{h_n^r \to f_n^r}\Big) = \mathcal{N}\Big(f_n^r \Big| \mu^t_{f_n^r}, v^t_{f_n^r}\Big), \\
\mu^t_{f_n^r} &= v^t_{f_n^r}\Big(\mu^{t-1}_{g^r \to f_n^r}/v^{t-1}_{g^r \to f_n^r} + \mu^{t-1}_{h_n^r \to f_n^r}/v^{t-1}_{h_n^r \to f_n^r}\Big), \\
v^t_{f_n^r} &= \Big(1/v^{t-1}_{g^r \to f_n^r} + 1/v^{t-1}_{h_n^r \to f_n^r}\Big)^{-1}.
\end{aligned}$$

Step 3: Compute the message from f_n^r to the factor h_n^r

$$\begin{aligned}
m^t_{f_n^r \to h_n^r}(f_n^r) &= \frac{q^t(f_n^r)}{m^{t-1}_{h_n^r \to f_n^r}(f_n^r)} = \frac{\mathcal{N}\Big(f_n^r \Big| \mu^t_{f_n^r}, v^t_{f_n^r}\Big)}{\mathcal{N}\Big(f_n^r \Big| \mu^{t-1}_{h_n^r \to f_n^r}, v^{t-1}_{h_n^r \to f_n^r}\Big)} = \mathcal{N}\Big(f_n^r \Big| \mu^t_{f_n^r \to h_n^r}, v^t_{f_n^r \to h_n^r}\Big), \\
\mu^t_{f_n^r \to h_n^r} &= v^t_{f_n^r \to h_n^r}\Big(\mu^t_{f_n^r}/v^t_{f_n^r} - \mu^{t-1}_{h_n^r \to f_n^r}/v^{t-1}_{h_n^r \to f_n^r}\Big), \\
v^t_{f_n^r \to h_n^r} &= \Big(1/v^t_{f_n^r} - 1/v^{t-1}_{h_n^r \to f_n^r}\Big)^{-1}.
\end{aligned}$$

Step 4: Compute the message from the h_n^r factor to y_n^r

$$\begin{aligned}
m^t_{h_n^r \to y_n^r}(y_n^r) &= \int h_n^r(y_n^r, f_n^r)\, m^t_{f_n^r \to h_n^r}(f_n^r)\, df_n^r \\
&= \int \mathcal{N}\Big(y_n^r \Big| f_n^r, \beta_r\Big)\, \mathcal{N}\Big(f_n^r \Big| \mu^t_{f_n^r \to h_n^r}, v^t_{f_n^r \to h_n^r}\Big)\, df_n^r \\
&= \mathcal{N}\Big(y_n^r \Big| \mu^t_{f_n^r \to h_n^r}, v^t_{f_n^r \to h_n^r} + \beta_r\Big).
\end{aligned}$$

Step 5: Compute the (approximate) posterior on y_n^r as the product of all incoming messages

$$\begin{aligned}
q^t(y_n^r) &= m^t_{l_n^r \to y_n^r}(y_n^r)\, m^t_{h_n^r \to y_n^r}(y_n^r)\, m^{t-1}_{k_n \to y_n^r}(y_n^r) \\
&= \mathbb{I}(y_n^r > 0)\, \mathcal{N}\Big(y_n^r \Big| \mu^t_{h_n^r \to y_n^r}, v^t_{h_n^r \to y_n^r}\Big)\, \mathcal{N}\Big(y_n^r \Big| \mu^{t-1}_{k_n \to y_n^r}, v^{t-1}_{k_n \to y_n^r}\Big) \\
&= \mathbb{I}(y_n^r > 0)\, \mathcal{N}\Big(y_n^r \Big| \hat{\mu}^t_{y_n^r}, \hat{v}^t_{y_n^r}\Big),
\end{aligned}$$

where

$$\begin{aligned}
\hat{\mu}^t_{y_n^r} &= \hat{v}^t_{y_n^r}\Big(\mu^t_{h_n^r \to y_n^r}/v^t_{h_n^r \to y_n^r} + \mu^{t-1}_{k_n \to y_n^r}/v^{t-1}_{k_n \to y_n^r}\Big), \\
\hat{v}^t_{y_n^r} &= \Big(1/\mu^t_{h_n^r \to y_n^r} + 1/v^{t-1}_{k_n \to y_n^r}\Big)^{-1}.
\end{aligned}$$

The product $\mathbb{I}(y_n^r > 0)\mathcal{N}(y_n^r|\hat{\mu}^t_{y_n^r}, \hat{v}^t_{y_n^r})$ is approximated by moment matching (see Appendix C.5 for the derivation of the moments of a truncated Gaussian). The approximate posterior on y_n^r is then given by

$$\begin{aligned}
q^t(y_n^r) &= \mathbb{I}(y_n^r > 0)\mathcal{N}(y_n^r|\hat{\mu}^t_{y_n^r}, \hat{v}^t_{y_n^r}) \approx \mathcal{N}(y_n^r|\mu^t_{y_n^r}, v^t_{y_n^r}), \\
\mu^t_{y_n^r} &= \hat{\mu}^t_{y_n^r} + \sqrt{\hat{v}^t_{y_n^r}}\frac{\mathcal{N}(z_n)}{\Phi(z_n)}, \\
v^t_{y_n^r} &= \hat{v}^t_{y_n^r}\left(1 - z_n\frac{\mathcal{N}(z_n)}{\Phi(z_n)} - \left(\frac{\mathcal{N}(z_n)}{\Phi(z_n)}\right)^2\right),
\end{aligned}$$

where $z_n = \frac{\hat{\mu}^t_{y^r_n}}{\sqrt{\hat{v}^t_{y^r_n}}}$.

Step 6: Compute the message from y^r_n to the factor k_n

$$m^t_{y^r_n \to k_n}(y^r_n) = \frac{q^t(y^r_n)}{m^{t-1}_{k_n \to y^r_n}(y^r_n)} = \frac{\mathcal{N}\Big(y^r_n \Big| \mu^t_{y^r_n}, v^t_{y^r_n}\Big)}{\mathcal{N}\Big(y^r_n \Big| \mu^{t-1}_{k_n \to y^r_n}, v^{t-1}_{k_n \to y^r_n}\Big)} = \mathcal{N}\Big(y^r_n \Big| \mu^t_{y^r_n \to k_n}, v^t_{y^r_n \to k_n}\Big),$$

$$\mu^t_{y^r_n \to k_n} = v^t_{y^r_n \to k_n}\Big(\mu^t_{y^r_n}/v^t_{y^r_n} - \mu^{t-1}_{k_n \to y^r_n}/v^{t-1}_{k_n \to y^r_n}\Big),$$

$$v^t_{y^r_n \to k_n} = \Big(1/v^t_{y^r_n} - 1/v^{t-1}_{k_n \to y^r_n}\Big)^{-1}.$$

Step 7: Compute the message from the k_n factor to $y^{e_i}_n$

$$\begin{aligned} m^t_{k_n \to y^{e_i}_n}(y^{e_i}_n) &= \int k_n(y_i, y^r_n, \{y^{e_i}_n\}_{i=1}^{E_n})\, m^t_{y^r_n \to k_n}(y^r_n) \prod_{j \neq i} m^t_{y^{e_j}_n \to k_n}(y^{e_j}_n)\, dy^r_n\, d\{y^{e_j}_n\}_{j \neq i} \\ &= \int \mathcal{N}\Big(y^{e_i}_n \Big| y_n - y^r_n - \sum_{j \neq i}^{E_n} y^{e_j}_n, v\Big)\, \mathcal{N}\Big(y^r_n \Big| \mu^t_{y^r_n \to k_n}, v^t_{y^r_n \to k_n}\Big) \\ &\times \prod_{j \neq i}^{E_n} \mathcal{N}\Big(y^{e_j}_n \Big| \mu^t_{y^{e_j}_n \to k_n}, v^t_{y^{e_j}_n \to k_n}\Big)\, dy^r_n\, d\{y^{e_j}_n\}_{j \neq i} \\ &= \mathcal{N}\Big(y^{e_i}_n \Big| \mu^t_{k_n \to y^{e_i}_n}, v^t_{k_n \to y^{e_i}_n}\Big), \end{aligned}$$

where

$$\begin{aligned} \mu^t_{k_n \to y^{e_i}_n} &= y_n - \mu^t_{y^r_n \to k_n} - \textstyle\sum_{j \neq i} \mu^t_{y^{e_j}_n \to k_n}, \\ v^t_{k_n \to y^{e_i}_n} &= v + v^t_{y^r_n \to k_n} + \textstyle\sum_{j \neq i} v^t_{y^{e_j}_n \to k_n}. \end{aligned}$$

Step 8: Compute the (approximate) posterior on $y^{e_i}_n$

$$q^t(y^{e_i}_n) = m^t_{l^{e_i}_n \to y^{e_i}_n}(y^{e_i}_n)\, m^t_{h^{e_i}_n \to y^{e_i}_n}(y^{e_i}_n)\, m^t_{k_n \to y^{e_i}_n}(y^{e_i}_n).$$

This step is identical to step 5, but with the message $m^{t-1}_{k_n \to y^{e_i}_n}(y^{e_i}_n)$ replaced with the new (updated) message $m^t_{k_n \to y^{e_i}_n}(y^{e_i}_n)$.

Step 9: Compute the message from $y^{e_i}_n$ to the factor $h^{e_i}_n$

$$m^t_{y^{e_i}_n \to h^{e_i}_n}(y^{e_i}_n) = \frac{q^t(y^{e_i}_n)}{m^t_{h^{e_i}_n \to y^{e_i}_n}(y^{e_i}_n)} = \frac{\mathcal{N}(y^{e_i}_n | \mu^t_{y^{e_i}_n}, v^t_{y^{e_i}_n})}{\mathcal{N}\Big(y^{e_i}_n \Big| \mu^t_{h^{e_i}_n \to y^{e_i}_n}, v^{t-1}_{h^{e_i}_n \to y^{e_i}_n}\Big)} = \mathcal{N}\Big(y^{e_i}_n \Big| \mu^t_{y^{e_i}_n \to h^{e_i}_n}, v^t_{y^{e_i}_n \to h^{e_i}_n}\Big),$$

$$\mu^t_{y^{e_i}_n \to h^{e_i}_n} = v^t_{y^{e_i}_n \to h^{e_i}_n}\Big(\mu^t_{y^{e_i}_n}/v^t_{y^{e_i}_n} - \mu^t_{h^{e_i}_n \to y^{e_i}_n}/v^t_{h^{e_i}_n \to y^{e_i}_n}\Big),$$

$$v^t_{y^{e_i}_n \to h^{e_i}_n} = \Big(1/v^t_{y^{e_i}_n} - 1/v^t_{h^{e_i}_n \to y^{e_i}_n}\Big)^{-1}.$$

Step 10: Compute the message from the $h^{e_i}_n$ factor to $f^{e_i}_n$

$$\begin{aligned} m^t_{h^{e_i}_n \to f^{e_i}_n}(f^{e_i}_n) &= \int h^{e_i}_n(y^{e_i}_n, f^{e_i}_n)\, m^t_{y^{e_i}_n \to h^{e_i}_n}(y^{e_i}_n)\, dy^{e_i}_n \\ &= \int \mathcal{N}\Big(f^{e_i}_n \Big| y^{e_i}_n, \beta_e\Big)\, \mathcal{N}\Big(y^{e_i}_n \Big| \mu^t_{y^{e_i}_n \to h^{e_i}_n}, v^t_{y^{e_i}_n \to h^{e_i}_n}\Big)\, dy^{e_i}_n \qquad \text{(C.17)} \\ &= \mathcal{N}\Big(f^{e_i}_n \Big| \mu^t_{y^{e_i}_n \to h^{e_i}_n}, v^t_{y^{e_i}_n \to h^{e_i}_n} + \beta_e\Big). \end{aligned}$$

Step 11: Compute the (new) posterior on $f_n^{e_i}$

$$q^t(f_n^{e_i}) = m^t_{g^e \to f_n^{e_i}}(f_n^{e_i})\, m^t_{h_n^{e_i} \to f_n^{e_i}}(f_n^{e_i}).$$

This step is identical to step 2, but uses the new (updated) message at time t: $m^t_{h_n^{e_i} \to f_n^{e_i}}(f_n^{e_i})$.

Step 12: Compute the message from $f_n^{e_i}$ to the factor g^e

$$m^t_{f_n^{e_i} \to g^e}(f_n^{e_i}) = \frac{q^t(f_n^{e_i})}{m^t_{g^e \to f_n^{e_i}}(f_n^{e_i})} = \frac{\mathcal{N}\Big(f_n^{e_i}\Big|\mu^t_{f_n^{e_i}}, v^t_{f_n^{e_i}}\Big)}{\mathcal{N}\Big(f_n^{e_i}\Big|\mu^t_{g^e \to f_n^{e_i}}, v^t_{g^e \to f_n^{e_i}}\Big)} = \mathcal{N}\Big(f_n^{e_i}\Big|\mu^t_{f_n^{e_i} \to g^e}, v^t_{f_n^{e_i} \to g^e}\Big),$$

$$\mu^t_{f_n^{e_i} \to g^e} = v^t_{f_n^{e_i} \to g^e}\Big(\mu^t_{f_n^{e_i}}/v^t_{f_n^{e_i}} - \mu^t_{g^e \to f_n^{e_i}}/v^t_{g^e \to f_n^{e_i}}\Big),$$

$$v^t_{f_n^{e_i} \to g^e} = \Big(1/v^t_{f_n^{e_i}} - 1/v^t_{g^e \to f_n^{e_i}}\Big)^{-1}.$$

These steps are then iterated until convergence. All the messages are initialized to be uniform Gaussians, i.e. zero-mean and infinite variance. Notice that messages above correspond only to a subset of the all messages passed, since an analogous message flow, in the opposite direction of the one presented, is required in order for EP to work.

C.4 Expectation propagation for BAM-LR

In this appendix, we derive an EP algorithm for performing approximate inference in a Bayesian additive model where the components are linear functions of the inputs. Fig. C.1 shows a factor graph representation of this model, where the notation in blue represents the steps of the EP algorithm.

For notational convenience, let us define the following factors:

$$\begin{aligned} g^r(\boldsymbol{\eta}_r) &= \mathcal{N}(\boldsymbol{\eta}_r|\mathbf{0}, \boldsymbol{\Sigma}_r^0) \\ g^e(\boldsymbol{\eta}_e) &= \mathcal{N}(\boldsymbol{\eta}_e|\mathbf{0}, \boldsymbol{\Sigma}_e^0) \\ h_n^r(y_n^r, , \mathbf{x}_n^r, \boldsymbol{\eta}_r) &= \mathcal{N}(y_n^r|(\mathbf{x}_n^r)^{\mathrm{T}}\boldsymbol{\eta}_r, \beta_r) \\ h_n^{e_i}(y_n^{e_i}, \mathbf{x}_n^{e_i}, \boldsymbol{\eta}_e) &= \mathcal{N}(y_n^{e_i}|(\mathbf{x}_n^{e_i})^{\mathrm{T}}\boldsymbol{\eta}_e, \beta_e). \end{aligned}$$

The remaining factors are the same from Appendix C.3. Hence, we only need to revisit the messages that involve these new factors. The steps 4–8 are then similar to the model with GP components and correspond to steps 5–9, respectively, of the EP algorithm described in Appendix C.3. The remaining new steps are the following:

Step 1: Compute the posteriors over the routine weights $q^t(\boldsymbol{\eta}_r)$ as the the product of all incoming messages at time $t-1$. For both computational and numerical stability reasons, we parameterize the posterior on $\boldsymbol{\eta}_r$ and the Gaussian messages that involve it by their, by their natural parameters: the precision matrix $\boldsymbol{\Lambda}$ and

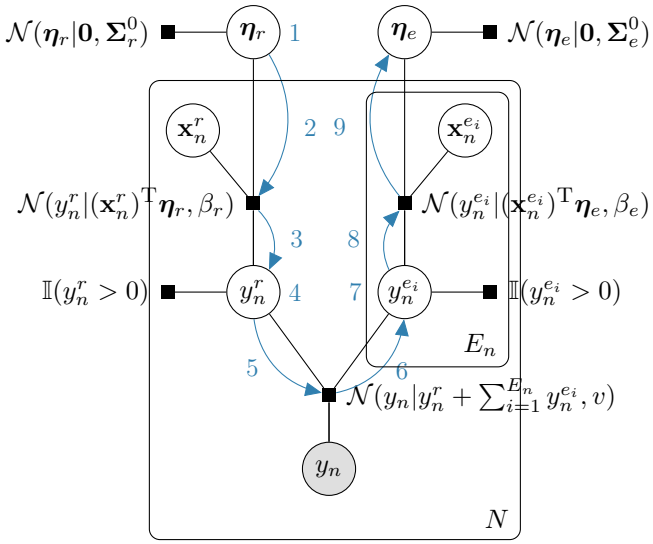


Figure C.1: Factor graph of the proposed Bayesian additive model with linear components. The blue arrows represent the message-passing algorithm for performing approximate Bayesian inference. The second flow of messages starting from the weights factor for the events component that goes in the opposite direction is not shown.

the precision-adjusted mean vector $\boldsymbol{\theta}$. The posterior then becomes

$$q^t(\boldsymbol{\eta}_r) = \mathcal{N}(\boldsymbol{\eta}_r|\mathbf{0}, \boldsymbol{\Sigma}_r^0) \prod_{n=1}^{N} m_{h_n^r \to \boldsymbol{\eta}_r}^{t-1}(\boldsymbol{\eta}_r) = \mathcal{N}(\boldsymbol{\eta}_r|\boldsymbol{\theta}_{\boldsymbol{\eta}_r}^t, \boldsymbol{\Lambda}_{\boldsymbol{\eta}_r}^t),$$

$$\boldsymbol{\theta}_{\boldsymbol{\eta}_r}^t = \sum_{n=1}^{N} \boldsymbol{\theta}_{h_n^r \to \boldsymbol{\eta}_r}^{t-1},$$

$$\boldsymbol{\Lambda}_{\boldsymbol{\eta}_r}^t = \boldsymbol{\Lambda}_r^0 + \sum_{n=1}^{N} \boldsymbol{\Lambda}_{h_n^r \to \boldsymbol{\eta}_r}^{t-1}.$$

Step 2: Compute the message from $\boldsymbol{\eta}_r$ to the factor h_n^r

$$m_{\boldsymbol{\eta}_r \to h_n^r}^t(\boldsymbol{\eta}_r) = \frac{q^t(\boldsymbol{\eta}_r)}{m_{h_n^r \to \boldsymbol{\eta}_r}^{t-1}(\boldsymbol{\eta}_r)} = \mathcal{N}\left(\boldsymbol{\eta}_r \middle| \boldsymbol{\theta}_{\boldsymbol{\eta}_r \to h_n^r}^t, \boldsymbol{\Lambda}_{\boldsymbol{\eta}_r \to h_n^r}^t\right),$$

$$\boldsymbol{\theta}_{\boldsymbol{\eta}_r \to h_n^r}^t = \boldsymbol{\theta}_{\boldsymbol{\eta}_r}^t - \boldsymbol{\theta}_{h_n^r \to \boldsymbol{\eta}_r}^{t-1},$$

$$\boldsymbol{\Lambda}_{\boldsymbol{\eta}_r \to h_n^r}^t = \boldsymbol{\Lambda}_{\boldsymbol{\eta}_r}^t - \boldsymbol{\Lambda}_{h_n^r \to \boldsymbol{\eta}_r}^{t-1}.$$

Step 3: Compute the message from the h_n^r factor to y_n^r

$$m_{h_n^r \to y_n^r}^t(y_n^r) = \int h_n^r(y_n^r, \mathbf{x}_n^r, \boldsymbol{\eta}_r)\, m_{\boldsymbol{\eta}_r \to h_n^r}^t(\boldsymbol{\eta}_r)\, d\boldsymbol{\eta}_r$$

$$= \mathcal{N}\left(y_n^r \middle| \mu_{h_n^r \to y_n^r}^t, v_{h_n^r \to y_n^r}^t\right),$$

$$\mu_{h_n^r \to y_n^r}^t = (\mathbf{x}_n^r)^{\mathrm{T}} (\boldsymbol{\Lambda}_{\boldsymbol{\eta}_r \to h_n^r}^t)^{-1} \boldsymbol{\theta}_{\boldsymbol{\eta}_r \to h_n^r}^t,$$

$$v_{h_n^r \to y_n^r}^t = (\mathbf{x}_n^r)^{\mathrm{T}} (\boldsymbol{\Lambda}_{\boldsymbol{\eta}_r \to h_n^r}^t)^{-1} \mathbf{x}_n^r + \beta_r.$$

Step 9: Compute the message from the $h_n^{e_i}$ factor to $\boldsymbol{\eta}_e$

$$\begin{aligned} m^t_{h_n^{e_i}\to\boldsymbol{\eta}_e}(\boldsymbol{\eta}_e) &= \int h_n^{e_i}(y_n^{e_i}, \mathbf{x}_n^{e_i}, \boldsymbol{\eta}_e)\, m^t_{y_n^{e_i}\to h_n^{e_i}}(y_n^{e_i})\, dy_n^{e_i} \\ &= \mathcal{N}\Big(\boldsymbol{\eta}_e \Big| \mu^t_{h_n^{e_i}\to\boldsymbol{\eta}_e}, v^t_{h_n^{e_i}\to\boldsymbol{\eta}_e}\Big), \\ \boldsymbol{\theta}^t_{h_n^{e_i}\to\boldsymbol{\eta}_e} &= \boldsymbol{\Lambda}^t_{h_n^{e_i}\to\boldsymbol{\eta}_e}(\mathbf{x}_n^{e_i})^{-\mathrm{T}}\mu^t_{y_n^{e_i}\to h_n^{e_i}}, \\ \boldsymbol{\Lambda}^t_{h_n^{e_i}\to\boldsymbol{\eta}_e} &= \Big((\mathbf{x}_n^{e_i})^{-\mathrm{T}}(v^t_{y_n^{e_i}\to h_n^{e_i}} + \beta_e)(\mathbf{x}_n^{e_i})^{-1}\Big)^{-1}. \end{aligned}$$

These steps are then iterated until convergence. All the messages are initialized to be uniform Gaussians, i.e. zero-mean and infinite variance. As with the message-passing algorithm for model with GP components, there is a symmetric flow of messages in the opposite direction of the ones described whose equations are identical to the ones presented.

C.5 Moments of a one-side truncated Gaussian

In order to make this result more general, we will derive these moments using a general Gaussian distribution $\mathcal{N}(x|\mu,\sigma^2)$ and a lower threshold l. The normalization constant, Z, of this one-side truncated Gaussian is given by

$$\begin{aligned} Z &= \int \mathbb{I}(x > l)\, \mathcal{N}(x|\mu,\sigma^2)\, dx \\ &= \int_l^{+\infty} \mathcal{N}(x|\mu,\sigma^2)\, dx = \Phi\Big(\frac{\mu - l}{\sigma}\Big), \end{aligned} \tag{C.18}$$

where $\Phi(a)$ denotes the value of the cumulative distribution functions (CDF) of a Gaussian distribution evaluated at a. Differentiating both sides w.r.t. μ gives

$$\begin{aligned} &\int_l^{+\infty} \frac{\partial \mathcal{N}(x|\mu,\sigma^2)}{\partial \mu}\, dx = \frac{\partial\Big(\Phi(\frac{\mu-l}{\sigma})\Big)}{\partial\mu} \\ \Leftrightarrow &\int_l^{+\infty} \Big(\frac{x-\mu}{\sigma^2}\Big)\, \mathcal{N}(x|\mu,\sigma^2)\, dx = \frac{1}{\sigma}\mathcal{N}\Big(\frac{\mu-l}{\sigma}\Big), \end{aligned}$$

where we made use of the fact that $\frac{\partial\Phi(z)}{\partial\mu} = \mathcal{N}(z)\frac{\partial z}{\partial\mu}$. Continuing developing the expression gives

$$\begin{aligned} &\Leftrightarrow \frac{1}{\sigma^2}\int_l^{+\infty} x\,\mathcal{N}(x|\mu,\sigma^2)\, dx - \frac{\mu}{\sigma^2}\int_l^{+\infty} \mathcal{N}(x|\mu,\sigma^2)\, dx = \frac{1}{\sigma}\mathcal{N}\Big(\frac{\mu-l}{\sigma}\Big) \\ &\Leftrightarrow \int_l^{+\infty} x\,\mathcal{N}(x|\mu,\sigma^2)\, dx - \mu\int_l^{+\infty} \mathcal{N}(x|\mu,\sigma^2)\, dx = \sigma\,\mathcal{N}\Big(\frac{\mu-l}{\sigma}\Big) \\ &\Leftrightarrow \underbrace{\int_l^{+\infty} x\,\mathcal{N}(x|\mu,\sigma^2)\, dx}_{=Z\cdot\mathbb{E}[x]} - \mu\,\underbrace{\Phi\Big(\frac{\mu-l}{\sigma}\Big)}_{=Z} = \sigma\,\mathcal{N}\Big(\frac{\mu-l}{\sigma}\Big) \\ &\Leftrightarrow \mathbb{E}[x]\,\Phi\Big(\frac{\mu-l}{\sigma}\Big) - \mu\,\Phi\Big(\frac{\mu-l}{\sigma}\Big) = \sigma\,\mathcal{N}\Big(\frac{\mu-l}{\sigma}\Big) \\ &\Leftrightarrow \mathbb{E}[x] = \mu + \sigma\,\frac{\mathcal{N}(\frac{\mu-l}{\sigma})}{\Phi(\frac{\mu-l}{\sigma})}. \end{aligned} \tag{C.19}$$

In order to determine the second moment, we start by differentiating both sides of (C.18) twice w.r.t. μ

$$\begin{aligned}
&\int_l^{+\infty} \frac{\partial^2 \mathcal{N}(x|\mu,\sigma^2)}{\partial \mu^2}\, dx = \frac{1}{\sigma}\frac{\partial \mathcal{N}(\frac{\mu-l}{\sigma})}{\partial \mu} \\
\Leftrightarrow &\int_l^{+\infty} \frac{\partial^2 \mathcal{N}(x|\mu,\sigma^2)}{\partial \mu^2}\, dx = -\frac{1}{\sigma}\Big(\frac{\mu-l}{\sigma^2}\Big)\,\mathcal{N}\Big(\frac{\mu-l}{\sigma}\Big),
\end{aligned} \tag{C.20}$$

where we made use of the fact that

$$\frac{\partial \mathcal{N}(x|\mu,\sigma^2)}{\partial \mu} = -(\frac{x-\mu}{\sigma^2})\,\mathcal{N}(x|\mu,\sigma^2). \tag{C.21}$$

Continuing differentiating (C.20), we get

$$\begin{aligned}
&\Leftrightarrow \int_l^{+\infty} \frac{x^2-2\mu x+\mu^2-\sigma^2}{\sigma^4}\mathcal{N}(x|\mu,\sigma^2)\, dx = -\frac{1}{\sigma}\frac{\mu-l}{\sigma^2}\,\mathcal{N}\Big(\frac{\mu-l}{\sigma}\Big) \\
&\Leftrightarrow \int_l^{+\infty} (x^2-2\mu x+\mu^2-\sigma^2)\mathcal{N}(x|\mu,\sigma^2)\, dx = -\sigma(\mu-l)\mathcal{N}\Big(\frac{\mu-l}{\sigma}\Big) \\
&\Leftrightarrow \underbrace{\int_l^{+\infty} x^2\mathcal{N}(x|\mu,\sigma^2)\, dx}_{=Z\,\mathbb{E}[x^2]} - 2\mu \underbrace{\int_l^{+\infty} x\,\mathcal{N}(x|\mu,\sigma^2) dx}_{=Z\,\mathbb{E}[x]} \\
&\quad + (\mu^2-\sigma^2)\underbrace{\int_l^{+\infty} \mathcal{N}(x|\mu,\sigma^2) dx}_{=Z} = -\sigma(\mu-l)\,\mathcal{N}\Big(\frac{\mu-l}{\sigma}\Big) \\
&\Leftrightarrow \mathbb{E}[x^2]\,Z - 2\mu\,\mathbb{E}[x]\,Z + (\mu^2-\sigma^2)\,Z = -\sigma\,(\mu-l)\,\mathcal{N}\Big(\frac{\mu-l}{\sigma}\Big) \\
&\Leftrightarrow \mathbb{E}[x^2] - 2\mu\,\mathbb{E}[x] + \mu^2 - \sigma^2 = -\sigma\,(\mu-l)\,\frac{\mathcal{N}(\frac{\mu-l}{\sigma})}{\Phi(\frac{\mu-l}{\sigma})} \\
&\Leftrightarrow \mathbb{E}[x^2] - 2\mu\Big(\mu + \sigma\,\frac{\mathcal{N}(\frac{\mu-l}{\sigma})}{\Phi(\frac{\mu-l}{\sigma})}\Big) + \mu^2 - \sigma^2 = -\sigma\,(\mu-l)\,\frac{\mathcal{N}(\frac{\mu-l}{\sigma})}{\Phi(\frac{\mu-l}{\sigma})} \\
&\Leftrightarrow \mathbb{E}[x^2] - \mu^2 + 2\mu\sigma\,\frac{\mathcal{N}(\frac{\mu-l}{\sigma})}{\Phi(\frac{\mu-l}{\sigma})} - \sigma^2 = -\sigma\,(\mu-l)\,\frac{\mathcal{N}(\frac{\mu-l}{\sigma})}{\Phi(\frac{\mu-l}{\sigma})} \\
&\Leftrightarrow \mathbb{E}[x^2] = \mu^2 + \sigma^2 - \sigma\,(\mu-l)\,\frac{\mathcal{N}(\frac{\mu-l}{\sigma})}{\Phi(\frac{\mu-l}{\sigma})} - 2\mu\sigma\,\frac{\mathcal{N}(\frac{\mu-l}{\sigma})}{\Phi(\frac{\mu-l}{\sigma})} \\
&\Leftrightarrow \mathbb{E}[x^2] = \mu^2 + \sigma^2\Big(1 - \frac{(\mu-l)}{\sigma}\frac{\mathcal{N}(\frac{\mu-l}{\sigma})}{\Phi(\frac{\mu-l}{\sigma})}\Big) - 2\mu\sigma\,\frac{\mathcal{N}(\frac{\mu-l}{\sigma})}{\Phi(\frac{\mu-l}{\sigma})}.
\end{aligned} \tag{C.22}$$

We can now make use of the two first moments in order to determine the variance, which is given by

$$
\begin{aligned}
\mathbb{V}[x] &= \mathbb{E}[x^2] - \mathbb{E}[x]^2 \\
&= \mu^2 + \sigma^2\left(1 - \frac{(\mu - l)}{\sigma}\frac{\mathcal{N}(\frac{\mu-l}{\sigma})}{\Phi(\frac{\mu-l}{\sigma})}\right) - 2\mu\sigma\,\frac{\mathcal{N}(\frac{\mu-l}{\sigma})}{\Phi(\frac{\mu-l}{\sigma})} - \left(\mu + \sigma\,\frac{\mathcal{N}(\frac{\mu-l}{\sigma})}{\Phi(\frac{\mu-l}{\sigma})}\right)^2 \\
&= \mu^2 + \sigma^2 - \sigma(\mu - l)\,\frac{\mathcal{N}(\frac{\mu-l}{\sigma})}{\Phi(\frac{\mu-l}{\sigma})} - 2\mu\sigma\,\frac{\mathcal{N}(\frac{\mu-l}{\sigma})}{\Phi(\frac{\mu-l}{\sigma})} - \mu^2 + 2\mu\sigma\,\frac{\mathcal{N}(\frac{\mu-l}{\sigma})}{\Phi(\frac{\mu-l}{\sigma})} - \sigma^2\left(\frac{\mathcal{N}(\frac{\mu-l}{\sigma})}{\Phi(\frac{\mu-l}{\sigma})}\right)^2 \\
&= \sigma^2 - \sigma(\mu - l)\,\frac{\mathcal{N}(\frac{\mu-l}{\sigma})}{\Phi(\frac{\mu-l}{\sigma})} - \sigma^2\left(\frac{\mathcal{N}(\frac{\mu-l}{\sigma})}{\Phi(\frac{\mu-l}{\sigma})}\right)^2 \\
&= \sigma^2\left(1 - \frac{(\mu - l)}{\sigma}\frac{\mathcal{N}(\frac{\mu-l}{\sigma})}{\Phi(\frac{\mu-l}{\sigma})} - \left(\frac{\mathcal{N}(\frac{\mu-l}{\sigma})}{\Phi(\frac{\mu-l}{\sigma})}\right)^2\right). \qquad \text{(C.23)}
\end{aligned}
$$

Bibliography

Adams, B. and Kapan, D. (2009). Man bites mosquito: Understanding the contribution of human movement to vector-borne disease dynamics. *PLoS ONE*, 4(8):63–67.

Airoldi, E., Blei, D., Fienberg, S., and Xing, E. (2007). Combining stochastic block models and mixed membership for statistical network analysis. In *Statistical Network Analysis: Models, Issues, and New Directions*, pages 57–74. Springer.

Allen, J., Pertea, M., and Salzberg, S. (2004). Computational gene prediction using multiple sources of evidence. *Genome Research*, 14(1):142–148.

Allen, J. and Salzberg, S. (2005). JIGSAW: Integration of multiple sources of evidence for gene prediction. *Bioinformatics*, 21(18):3596–3603.

Andrieu, C., Freitas, N. D., Doucet, A., and Jordan, M. (2003). An introduction to mcmc for machine learning. *Machine learning*, 50(1-2):5–43.

Aucouturier, J. and Pachet, F. (2003). Representing musical genre: A state of the art. *Journal of New Music Research*, 32(1):83–93.

Bachrach, Y., Graepel, T., Minka, T., and Guiver, J. (2012). How to grade a test without knowing the answers - a Bayesian graphical model for adaptive Crowdsourcing and aptitude testing. In *Proceedings of the International Conference on Machine Learning*.

Barber, D. (2012). *Bayesian reasoning and machine learning*. Cambridge University Press.

Bellare, K. and Mccallum, A. (2007). Learning Extractors from Unlabeled Text using Relevant Databases. In *International Workshop on Information Integration on the Web*.

Bernardo, J., Bayarri, M., Berger, J., Dawid, A., Heckerman, D., Smith, A., and West, M. (2003). The variational bayesian em algorithm for incomplete data: with application to scoring graphical model structures. *Bayesian statistics*, 7:453–464.

Bernardo, J. and Smith, A. (2009). *Bayesian theory*, volume 405. John Wiley & Sons.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Blei, D. (2012). Probabilistic topic models. *Communications of the ACM*, 55(4):77–84.

Blei, D., Ng, A., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Bolte, F. (2006). Transport policy objectives: Traffic management as suitable tool. Technical report, Federal Highway Research Institute (BASt), Bergisch-Gladbach, Germany.

Borg, I. and Groenen, P. (2005). *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media.

Bosch, A., Zisserman, A., and Muñoz, X. (2006). Scene classification via plsa. In *Proceedings of the European Conference on Computer Vision*, pages 517–530. Springer.

Calabrese, F., Pereira, F., Giusy, L., Liu, L., and Ratti, C. (2010). The geography of taste: analyzing cell-phone mobility and social events. *Pervasive Computing*, 6030:22–37.

Callison-Burch, C. (2009). Fast, cheap, and creative: evaluating translation quality using amazon's mechanical turk. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 286–295. ACL.

Callison-Burch, C. and Dredze, M. (2010). Creating speech and language data with amazon's mechanical turk. In *Proceedings of the NAACL-HLT Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 1–12.

Chen, X., Lin, Q., and Zhou, D. (2013). Optimistic knowledge gradient policy for optimal budget allocation in crowdsourcing. In *Proceedings of the International Conference on Machine Learning*, pages 64–72.

Chipman, H., George, E., and McCulloch, R. (2010). Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, pages 266–298.

Dawid, A. P. and Skene, A. M. (1979). Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society: Series C*, 28(1):20–28.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B*, pages 1–38.

Donmez, P. and Carbonell, J. (2008). Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In *Proceedings of the ACM Conference on Information and Knowledge Management*, pages 619–628.

Donmez, P., Schneider, J., and Carbonell, J. (2010). A probabilistic framework to learn from multiple annotators with time-varying accuracy. In *Proceedings of the SIAM International Conference on Data Mining*, pages 826–837.

Dredze, M., Talukdar, P., and Crammer, K. (2009). Sequence learning from data with multiple labels. In *ECML-PKDD Workshop on Learning from Multi-Label Data*.

Duda, R. and Hart, P. (1973). *Pattern classification and scene analysis*, volume 3. Wiley.

Duvenaud, D., Nickisch, H., and Rasmussen, C. (2011). Additive gaussian processes. In *Advances in Neural Information Processing Systems*, pages 226–234.

Erosheva, E., Fienberg, S., and Joutard, C. (2007). Describing disability through individual-level mixture models for multivariate binary data. *The Annals of Applied Statistics*, 1(2):346.

Fei-Fei, L. and Perona, P. (2005). A Bayesian hierarchical model for learning natural scene categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 524–531. IEEE.

Fernandes, E. and Brefeld, U. (2011). Learning from partially annotated sequences. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 407–422.

(FHWA), F. H. A. (2006). Planned special events: Checklists for practitioners.

Gilks, W. (2005). *Markov chain Monte Carlo*. Wiley Online Library.

Gonzalez, M., Hidalgo, C., and Barabasi, A. (2008). Understanding individual human mobility patterns. *Nature*, 453(7196):779–782.

Groot, P., Birlutiu, A., and Heskes, T. (2011). Learning from multiple annotators with Gaussian processes. In *Proceedings of the International Conference on Artificial Neural Networks*, volume 6792, pages 159–164.

Hastie, T. and Tibshirani, R. (1990). *Generalized additive models*, volume 43. CRC Press.

Hastie, T., Tibshirani, R., and Friedman, J. (2003). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

Hoffman, M., Blei, D., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347.

Howe, J. (2008). *Crowdsourcing: why the power of the Crowd is driving the future of business*. Crown Publishing Group, 1 edition.

Jensen, F. (1996). *An introduction to Bayesian networks*, volume 210. UCL Press London.

Jiang, S., Fiore, G., Yang, Y., Jr, J. F., Frazzoli, E., and González, M. (2013). A review of urban computing for mobile phone traces: current methods, challenges and opportunities. In *Proceedings of the SIGKDD International Workshop on Urban Computing*, page 2. ACM.

Jordan, M. I., Ghahramani, Z., Jaakkola, T., and Saul, L. (1999). An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233.

Kapoor, A., Grauman, K., Urtasun, R., and Darrell, T. (2007). Active learning with Gaussian processes for object categorization. In *Proceedings of the International Conference on Computer Vision*, pages 1–8.

Koller, D. and Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

Krygsman, S., Dijst, M., and Arentze, T. (2004). Multimodal public transport: an analysis of travel time elements and the interconnectivity ratio. *Transport Policy*, 11(3):265–275.

Kschischang, F., Frey, B., and Loeliger, H. (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519.

Kwon, J., Mauch, M., and Varaiya, P. (2006). Components of congestion: delay from incidents, special events, lane closures, weather, potential ramp metering gain, and excess demand. *Transportation Research Record*, 1959(1):84–91.

Lacoste-Julien, S., Sha, F., and Jordan, M. (2009). Disclda: Discriminative learning for dimensionality reduction and classification. In *Advances in Neural Information Processing Systems*, pages 897–904.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning*, pages 282–289.

Lang, K. (1995). Newsweeder: Learning to filter netnews. In *Proceedings of the International Conference on Machine Learning*, pages 331–339.

Lawrence, N. D., Seeger, M., and Herbrich, R. (2003). Fast sparse Gaussian process methods: the informative vector machine. In *Advances in Neural Information Processing Systems*, pages 609–616. MIT Press.

Laws, F., Scheible, C., and Schütze, M. (2011). Active learning with amazon mechanical turk. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1546–1556. ACL.

Lawson, N., Eustice, K., Perkowitz, M., and Yetisgen-Yildiz, M. (2010). Annotating large email datasets for named entity recognition with mechanical turk. In *Proceedings of the NAACL-HLT Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 71–79. ACL.

Lewis, D. (1997). Reuters-21578 text categorization test collection, distribution 1.0.

Liu, D. and Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45:503–528.

Lowe, D. (1999). Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1150–1157. IEEE.

MacKay, D. (2003). *Information Theory, Inference & Learning Algorithms*. Cambridge University Press.

Mauá, D. and Cozman, F. (2009). Representing and classifying user reviews.

Mcauliffe, J. and Blei, D. (2008). Supervised topic models. In *Advances in Neural Information Processing Systems*, pages 121–128.

Middleham, F. (2006). Dynamic traffic management. Technical report, Ministry of Transport, Public Works, and Water Management, Directorate-General of Public Works and Water Management, AVV Transport Research Centre, Rotterdam, Netherlands.

Mimno, D. and McCallum, A. (2008). Topic models conditioned on arbitrary features with dirichlet-multinomial regression. *Proceedings of the Conference on Uncertainty in Artificial Intelligence*.

Minka, T. (2001). Expectation Propagation for approximate Bayesian inference. In *Proceedings of the Conference in Uncertainty in Artificial Intelligence*, pages 362–369.

Minka, T., Winn, J., Guiver, J., and Knowles, D. (2012). Infer.NET 2.5. Microsoft Research Cambridge. http://research.microsoft.com/infernet.

Murphy, K. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.

Ng, A. (2004). Feature selection, l1 vs. l2 regularization, and rotational invariance. In *Proceedings of the International Conference on Machine Learning*, page 78. ACM.

Niebles, J., Wang, H., and Fei-Fei, L. (2008). Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, 79(3):299–318.

Nocedal, J. and Wright, S. (2006). *Numerical Optimization*. World Scientific.

Pang, B. and Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the Annual Meeting on Association for Computational Linguistics*, pages 115–124. ACL.

Parisi, G. (1988). Statistical field theory. *Frontiers in Physics, Addison-Wesley*.

Park, S. and Choi, S. (2008). Gaussian processes for source separation. In *Proceeding of the Conference on Acoustics, Speech and Signal Processing*, pages 1909–1912.

Pearl, J. (2014). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann.

Pereira, F., Rodrigues, F., Polisciuc, E., and Ben-Akiva, M. (2014a). Why so many people? explaining nonhabitual transport overcrowding with internet data. *Transactions on Intelligent Transportation Systems*, 16(3):1370–1379.

Pereira, F. C., Rodrigues, F., and Ben-Akiva, M. (2014b). Using data from the web to predict public transport arrivals under special events scenarios. *Journal of Intelligent Transportation Systems*, 19(3):273–288.

Potier, F., Bovy, P., and Liaudat, C. (2003). Big events: planning, mobility management. In *Proceedings of the European Transport Conference*.

Rabiner, L. R. (1990). A tutorial on hidden markov models and selected applications in speech recognition. In *Readings in Speech Recognition*, pages 267–296. Morgan Kaufmann Publishers Inc.

Rabinovich, M. and Blei, D. (2014). The inverse regression topic model. In *Proceedings of the International Conference on Machine Learning*, pages 199–207.

Ramage, D., Hall, D., Nallapati, R., and Manning, C. (2009). Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 248–256. ACL.

Ramshaw, L. and Marcus, M. (1995). Text chunking using transformation-based learning. In *Proceedings of the Workshop on Very Large Corpora*, pages 82–94. ACL.

Rasmussen, C. E. and Williams, C. (2005). *Gaussian processes for machine learning (Adaptive computation and machine learning)*. MIT Press.

Ravikumar, P., Lafferty, J., Liu, H., and Wasserman, L. (2009). Sparse additive models. *Journal of the Royal Statistical Society: Series B*, 71(5):1009–1030.

Raykar, V., Yu, S., Zhao, L., Jerebko, A., Florin, C., Valadez, G., Bogoni, L., and Moy, L. (2009). Supervised learning from multiple experts: whom to trust when everyone lies a bit. In *Proceedings of the International Conference on Machine Learning*, pages 889–896.

Raykar, V., Yu, S., Zhao, L., Valadez, G., Florin, C., Bogoni, L., and Moy, L. (2010). Learning from Crowds. *Journal of Machine Learning Research*, pages 1297–1322.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407.

Rodrigues, F., Borysov, S., Ribeiro, B., and Pereira, F. (2015a). A bayesian additive model for understanding public transport usage in special events. *Submitted for publication*.

Rodrigues, F., Lourenço, M., Pereira, F., and Ribeiro, B. (2015b). Learning supervised topic models from crowds. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*.

Rodrigues, F., Lourenço, M., Ribeiro, B., and Pereira, F. (2015c). Learning supervised topic models from crowds. *Submitted for publication*.

Rodrigues, F., Pereira, F., and Ribeiro, B. (2013a). Learning from multiple annotators: distinguishing good from random labelers. *Pattern Recognition Letters*, pages 1428–1436.

Rodrigues, F., Pereira, F., and Ribeiro, B. (2013b). Sequence labeling with multiple annotators. *Machine Learning*, pages 1–17.

Rodrigues, F., Pereira, F., and Ribeiro, B. (2014). Gaussian process classification and active learning with multiple annotators. In *Proceedings of the International Conference on Machine Learning*, pages 433–441.

Russell, B., Torralba, A., Murphy, K., and Freeman, W. (2008). Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1-3):157–173.

Sang, E. and Meulder, F. D. (2003). Introduction to the conll-2003 shared task: language-independent named entity recognition. In *Proceedings of the Conference on Natural Language Learning at NAACL-HLT*, volume 4, pages 142–147.

Sheng, V., Provost, F., and Ipeirotis, P. (2008). Get another label? Improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining*, pages 614–622.

Smyth, P., Fayyad, U., Burl, M., Perona, P., and Baldi, P. (1995). Inferring ground truth from subjective labelling of venus images. In *Advances in Neural Information Processing Systems*, pages 1085–1092.

Snow, R., O'Connor, B., Jurafsky, D., and Ng, A. (2008). Cheap and fast - but is it good?: Evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 254–263.

Song, C., Qu, Z., Blumm, N., and Barabasi, A. (2010). Limits of predictability in human mobility. *Science*, 327(5968):1018–1021.

Steyvers, M. and Griffiths, T. (2007). Probabilistic topic models. *Handbook of Latent Semantic Analysis*, 427(7):424–440.

Surowiecki, J. (2004). *The Wisdom of Crowds : Why the many are smarter than the few and how collective wisdom shapes business, economies, societies, and nations.* Doubleday.

Sutton, C. (2012). An introduction to conditional random fields. *Foundations and Trends® in Machine Learning*, 4(4):267–373.

Sutton, C. and McCallum, A. (2006). *Introduction to Conditional Random Fields for Relational Learning.* MIT Press.

Taddy, M. (2013). Multinomial inverse regression for text analysis. *Journal of the American Statistical Association*, 108(503):755–770.

Tzanetakis, G. and Cook, P. (2002). Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302.

van Oort, N., Brands, T., and de Romph, E. (2015). Short term ridership prediction in public transport by processing smart card data. In *Proceedings of the Annual Meeting of the Transportation Research Board.*

Voyer, R., Nygaard, V., Fitzgerald, W., and Copperman, H. (2010). A hybrid model for annotating named entity training corpora. In *Proceedings of the Linguistic Annotation Workshop*, pages 243–246. ACL.

Wainwright, M. and Jordan, M. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305.

Wang, C., Blei, D., and Fei-Fei, L. (2009). Simultaneous image classification and annotation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1903–1910. IEEE.

Winkler, W. (1990). String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage.

Wu, O., Hu, W., and Gao, J. (2011). Learning to rank under multiple annotators. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1571–1576.

Xue, G., Li, Z., Zhu, H., and Liu, Y. (2009). Traffic-known urban vehicular route prediction based on partial mobility patterns. In *Proceedings of the International Conference on Parallel and Distributed Systems*, pages 369–375.

Yan, Y., Rosales, R., Fung, G., and Dy, J. (2011). Active learning from Crowds. In *Proceedings of the International Conference on Machine Learning*, pages 1161–1168.

Yan, Y., Rosales, R., Fung, G., Schmidt, M., Valadez, G., Bogoni, L., Moy, L., and Dy, J. (2010). Modeling annotator expertise: Learning when everybody knows a bit of something. *Journal of Machine Learning Research*, 9:932–939.

Zeno, S., Duvvuri, R., and Millard, R. (1995). *The educator's word frequency guide.* Touchstone Applied Science Associates.

Zhu, J., Ahmed, A., and Xing, E. (2012). Medlda: Maximum margin supervised topic models. *Journal of Machine Learning Research*, 13(1):2237–2278.