Image Dating, a Case Study to Evaluate the Inter-Battery Topic Model

JOHN PERTOFT
Image Dating, a Case Study to Evaluate the Inter-Battery Topic Model

JOHN PERTOFT

Master’s Thesis at CSC
Supervisors: Hedvig Kjellström, Cheng Zhang
Examiner: Danica Kragic
Abstract

The Inter-Battery Topic Model (IBTM) is an extension of the well known Latent Dirichlet Allocation (LDA) topic model. It gives a factorized representation of multimodal (in this case two views) data, which better separates variation in observed data that is present in both views from variation that is present only in one of the separate views. This thesis is an evaluation and application study of this model with the aim of showing how it can be used in the very difficult classification task of dating grayscale face portraits from a dataset collected from highschool yearbooks. This task has very high intra-class variation and low inter-class variation which calls for techniques to extract the necessary information. An online-trained model is also implemented and evaluated as well as a simplification of the model more suited for this data specifically.

The results show improved performance over LDA showing that the factorizing property of IBTM has a positive effect on performance for this type of classification task.
Referat

Bilddatering, en applikationsstudie för att evaluera Inter-Battery Topic Model

Inter-Battery Topic Model (IBTM) är en vidareutveckling av den välkända Latent Dirichlet Allocation (LDA) topic-modellen. Den ger en faktoriserad representation av multi-modal data som bättre separerar variation i datat som finns i båda datavyer från den som finns i de enskilda datavyerna. Det här examensarbetet är en evaluering och applikationsstudie av modellen, med mål att visa hur den kan användas i den mycket svåra klassificeringsuppgiften att datera svartvita bilder från ett dataset skapat från amerikanska highschool-årsboks fotografi. Denna klassificeringsuppgift har väldigt hög inom-klass variation samt väldigt låg mellan-klass variation vilket kräver bättre sätt att extrahera den nödvändiga information för bra klassificering. En onlinetränad variant av modellen implementeras och evalueras också, samt en modellvariant som är mer anpassad för just denna typ av data.

Resultaten visar bättre prestanda än LDA vilket visar att den faktoriserade representationen från IBTM har en positiv effekt på prestanda in en klassificeringsuppgift av den här typen.
## Contents

1 Introduction
1.1 Problem ................................................................. 1
1.2 Objective ............................................................. 2
1.3 Contribution and Delimitations ................................. 2
1.4 Outline of Thesis ................................................... 2
1.5 Ethical Considerations .............................................. 2
1.6 Relationship to Studies ........................................... 3

2 Background
2.1 Likelihood, Prior, and the Posterior Distribution ............ 5
2.2 Beta and Dirichlet Distributions .................................. 5
2.3 Bayesian Models .................................................... 6
2.4 Graphical Models ................................................... 10
2.5 Probabilistic Topic Models ......................................... 10
2.6 Multimodal Data ..................................................... 12
2.7 Image Features ....................................................... 13
   2.7.1 Handcrafted Features .......................................... 13
   2.7.2 Deep Convolutional Neural Network Extracted Features 13
2.8 Visual Words .......................................................... 13
2.9 Topic Modeling for Classification ............................... 14

3 Inter-Battery Topic Modeling ...................................... 15
3.1 Inference .................................................................. 17

4 Related work
4.1 Topic Modeling ....................................................... 19
4.2 Image Dating ........................................................ 22

5 Method
5.1 Data ........................................................................ 23
5.2 Document Creation .................................................. 23
   5.2.1 Using SIFT and Intensity Histogram Features .......... 24
   5.2.2 Using CNN Features .......................................... 24
5.3 Training IBTM ......................................................... 25
Chapter 1

Introduction

1.1 Problem

As Bengio et al. point out in [3], the performance of machine learning methods is very dependent on the type of representation of the data to which it is applied. Manually engineering a representation of the data (feature extraction) is hard work and requires domain knowledge which is not always available. If instead the representation can be learnt from the data, application of machine learning would be both easier and faster. Bengio et al. argue that it would also be great progress for artificial intelligence (AI) since an AI must fundamentally understand the world around us and to do that it must be able to learn to find a good representation that can disentangle the underlying explanatory factors in its input data.

One approach to this representation learning is topic modeling. Topic modeling algorithms learn a latent semantic representation of documents that can then be used for other machine learning tasks. Many different kinds of topic models have been developed, each with different advantages and disadvantages that can be used for different types of problems. In this thesis a newly developed topic model called Inter-Battery Topic Model (IBTM) [50] is used. In order to evaluate the capabilities of this model, it is applied to the problem of determining when a photograph of a person’s face was taken (as in year or decade). The dataset used contains highschool yearbook photos of students’ faces labelled with the year that the photo was taken. This data has high intra-class variation and low inter-class variation arising from the fact that people from the same year will still have vastly different appearances and that the general appearance of people today is not so different from people 100 years ago. The main point of this thesis is not to solve the problem, but instead to explore the capabilities and how IBTM works for this very difficult classification task. However, this type of metadata extraction images could also be of use for historians working with large sets of unlabeled photos containing faces.

The idea is that IBTM will be able to better find and isolate the variation in data that can sufficiently separate images from the different years or eras from one another in the representation space. This is because IBTM aims to find a factorized
representation where one factor tries to explain variation that is private to a class and another factor that tries to explain the variation that actually captures the significant information about a class.

1.2 Objective

The aim and objective of this project is to provide an evaluation of the performance and properties of IBTM in a difficult classification task. The results could be of interest to someone facing a similar problem and/or has an interest in learning more about the capabilities of topic models.

1.3 Contribution and Delimitations

This thesis provides an evaluation of the newly developed Inter-Battery Topic Model which hopefully can provide further insights in the strengths and weaknesses of the model by applying it in a difficult classification task where instances of different labels are extremely entangled.

As stated previously, it could also demonstrate a method that could be useful for historians working with massive sets of visual data containing faces.

The code for IBTM will not be written as a part of this thesis. Adaptations of the model will however be developed and tested.

1.4 Outline of Thesis

The rest of the thesis is structured as follows. Chapter 2 presents the relevant background theory needed to understand all parts of the model and method. Chapter 3 explains the IBTM model in detail. Chapter 4 places IBTM in a context by explaining previous work both in other topic models dealing with similar problems as well as other types of methods dealing with similar problems. Chapter 5 follows, explaining how IBTM is applied to this problem and the changes made to it to create the variants of it. Chapter 6 presents the experiments, the results as well as interpretations of results. Finally, Chapter 7 presents a summary of what has been done and the conclusions that could be drawn based on the experiments and evaluation of this model.

1.5 Ethical Considerations

Most applications of machine learning are in general a great benefit to society but for the sake of argument, consider a possible case in the future where an insurance company might strictly use a machine learning algorithm for computing the cost of insurance or a bank using another type machine learning algorithm to see if a mortgage application should be approved or not. These are decisions that can have
1.6. RELATIONSHIP TO STUDIES

a large effect on people’s lives and it is important to be aware of the limitations of the chosen machine learning algorithm. Depending on the type of algorithm used to generate a decision in these examples, the level of interpretability of the models decision might differ greatly. On one end of the spectrum, with many deep learning models, for example, it is often very hard or even impossible to explain why a decision was made, whereas with generative bayesian models it is usually more interpretable to some degree.

If these machine learning algorithms are used to make decisions where the data is very entangled and hard or impossible to interpret for a human then it might not be entirely ethically responsible to base such a decision on the machine learning algorithms’ answer alone. A machine learning system could pick up on some signal in the data which causes the end result to discriminate based on gender or race. An example of this was shown in [9] where Google Ads showed ads for a senior executive position for men with much higher probability than for women.

At the same time, the humans making these decisions today are also flawed but the question of accountability is important. A human might be forgiven for making a bad decision but a machine learning system might not get that luxury.

1.6 Relationship to Studies

This thesis ties together knowledge acquired in multiple courses at KTH. It requires knowledge of the basics in calculus (SF1625, SF1626) and statistics (SF1901), techniques from the many machine learning courses (DD2431, DD2432, DD2434, DD2427) as well as computer vision methods (DD2423, DD2427). The work done in this thesis should cover the learning outcomes and goals of the Autonomous Systems track of the Computer Science master’s programme.
Chapter 2

Background

This chapter presents the relevant background theory.

2.1 Likelihood, Prior, and the Posterior Distribution

Given that $\theta$ are the unknown variables or parameters of a statistical model and that $D$ is the observed data, Bayes theorem is defined and used in the following way. The prior is the distribution over the variables in $\theta$ without any observations of data, it expresses the beliefs about some variables without any evidence. The likelihood is the probability of some observed data given values of the variables in $\theta$. The evidence which is also known as marginal likelihood is the probability of observed data marginalised over all possible values of $\theta$.

$$p(\theta|D) = \frac{p(D|\theta) p(\theta)}{p(D)}$$  \hspace{1cm} (2.1)

2.2 Beta and Dirichlet Distributions

The Beta distribution \cite{15} is a continuous distribution over scalar values constrained to values between 0 and 1. It is parametrized by two positive real valued scalars, $a$ and $b$. The parameters $a$ and $b$ can be seen as the number of positive and negative outcomes of some random event with two outcomes. $\frac{a}{a+b}$ gives the expected value of the distribution and larger values of $a$ and $b$ gives a distribution with smaller variance. The Beta distribution is commonly used in Bayesian statistics as a prior distribution for a random variable describing a probability.

The Dirichlet distribution \cite{14} is a generalization of the Beta distribution. It describes a continuous multivariate probability distribution with the property that it describes a K dimensional vector $x$ where all elements are restricted to values between 0 and 1 and $\sum x_k = 1$. For this reason the Dirichlet distribution is commonly
used in Bayesian models as a prior distribution over multinomial distributions. The Dirichlet distribution is parametrized by the K dimensional vector $\alpha$ with $\alpha_k > 0$ which controls the shape of the distribution over the multivariates. Small values of $\alpha_k$ give a sparser multinomial distribution while larger values give a more uniform distribution. Commonly a symmetric $\alpha$ is used where all elements have the same value. A non symmetric $\alpha$ skews the center and shape of the probability mass.

Further useful properties of both the Beta and Dirichlet distributions are that they are conjugate priors to the Bernoulli and multinomial distributions, respectively. This means that the posterior distributions will also be Beta or Dirichlet distributions. This is useful because it means that the model will remain as the same type of distribution even after seeing data, but with updated parameters, which is algebraically convenient.

In this thesis, $\theta \sim \text{Dir}(\alpha)$ is the notation used for saying that $\theta$ is drawn from a Dirichlet distribution parametrized by $\alpha$. Analogously for the Beta distribution, $\rho \sim \text{Beta}(a, b)$.

### 2.3 Bayesian Models

Data used in machine learning is often stochastic by itself or noisily or partially observed. This introduces plenty of uncertainty to the system and it is often beneficial to include the uncertainty in the model. The Bayesian way of approaching this is by defining the model such that the variables and parameters, where this uncertainty should be expressed, are stochastic variables. This means that a prior probability distribution is placed on these variables, $\theta$. The objective is then to infer the posterior distribution over these variables given the observed data $D$. The posterior distribution gives a measure of the uncertainty over the sought parameters through its variance and covariance.

Another useful property of Bayesian modeling is that of model averaging which makes this type of models less susceptible to overfitting. This was the theme of D. MacKay’s thesis [29] and he described how Bayes rule has a built in Occam’s razor, meaning unnecessarily complex models will be penalised giving, in essence, built in regularization. This was also discussed by Murray and Ghahramani [32].

The Bayesian way of computing (or approximating) the posterior distribution over the model parameters $\theta$ is contrasted by methods like Maximum Likelihood (ML) and Maximum a Posteriori (MAP) which instead give point estimates of $\theta$.

$$\theta_{ML} = \arg \max_{\theta} p(D|\theta)$$ (2.2)

$$\theta_{MAP} = \arg \max_{\theta} p(\theta|D) = \arg \max_{\theta} p(D|\theta)p(\theta)$$ (2.3)

To illustrate the advantages of modeling the uncertainty, consider a posterior distribution over $\theta$ with one high thin peak and one lower but wider part as in Figure 2.1. MAP estimation would give a point estimate which corresponds to the
2.3. BAYESIAN MODELS

Figure 2.1: Bimodal posterior distribution.

...top of the thin peak. If this estimation would then be used for prediction, the performance would be poor because most of the probability mass is in the lower but wider part of the probability density function. By instead using Bayesian methods to model uncertainty the prediction can use the uncertainty to make a more informed choice.

Many Bayesian models are generative models which makes the assumption that the observed data has been generated by an imaginary so called generative process. The generative process explains in what way the model assumes that some observed data came to be and is an easy way to illustrate how the models work. The generative process is thus commonly used when explaining probabilistic topic models which are Bayesian models. Defining a model by starting from a generative process has the advantage that it allows for separation of defining the hidden structure that represents what is of interest, as well as the assumptions made about the data from the inference of this structure. This makes for high interpretability. The task of trying to reverse this process by, given observed data, trying to find the posterior distribution over these latent variables is called inference and is the main computational task in these types of models.

Graphical models, explained in Section 2.4, are another way to illustrate how variables in a probabilistic model depend on each other. Graphical models are also commonly used to explain topic models.

The biggest downside of using Bayesian models is the increased complexity of computation when inferring the posterior distribution of the wanted latent variables. Compared to ML and MAP the increased complexity comes from having to deal with the evidence \( p(D) \) which is the normalizing factor in Bayes’ rule. For continuous parameter values the evidence is computed as an integral

\[
p(D) = \int_{\theta} p(D|\theta)p(\theta)d\theta
\]  

In fact, most Bayesian models actually lead to an analytically intractable integral because the latent variables might be coupled in a way that would make the computation complexity grow exponentially. To circumvent this, approximation methods are used. These methods fall into two different categories, sampling methods and variational methods, with different advantages and disadvantages. The advantage of using a sampling method is that it approaches the true posterior dis-
CHAPTER 2. BACKGROUND

Distribution when the number of samples goes to infinity but the downside is that it is computationally expensive. Many sampling methods are based on Markov Chain Monte Carlo (MCMC), which is a method based on constructing a markov chain on the latent variables \( \theta \), where the stationary distribution of the markov chain is the posterior distribution of \( \theta \).

The variational inference methods, on the other hand, are faster in the sense that they will likely give a better result than a sampling method, given the same time. There are different variants of variational inference, but the common idea is that the problem is transformed to an optimization problem by picking a family of distributions \( q(\theta) \) over the latent variables with new variational parameters and then optimizing these variational parameters to find a distribution \( q(\theta) \) that is close to the true posterior \( p(\theta|D) \). The closeness to the true posterior is measured with the Kullback-Leibler (KL) divergence, which is a measure of difference between probability distributions. The \( q \) that minimizes the KL divergence is then used as the approximated posterior distribution when doing predictions or using the distribution over the latent variables of the model. KL divergence is not symmetric meaning \( KL(q||p) \neq KL(p||q) \) and for variational inference it is defined as

\[
KL(q||p) = E_q \left[ \log \frac{q(\theta)}{p(\theta|D)} \right] = \int q(\theta) \log \frac{q(\theta)}{p(\theta|D)} d\theta
\]

Continuous case (2.5)

where \( q(\theta) \) is the variational distribution and \( p(\theta|D) \) is the true posterior distribution.

When using the KL divergence in this order (reverse KL divergence), the KL divergence will be low when \( q \) and \( p \) are both high, it will be high when \( q \) is high and \( p \) is low, and when \( q \) is low the KL divergence is always low because it is an expectation over \( q \). The intuition in this is that the KL divergence is minimized with a \( q \) that has its probability mass covering one of the modes in a possible multimodal distribution \( p \). This is illustrated in Figure 2.2a where the blue curves are the contours of the true posterior \( p \) and the red curves are the contours of the approximated distribution \( q \). The opposite order (forward KL divergence) would instead find a \( q \) that covers all modes as shown in Figure 2.2b.

Since the KL divergence contains the true posterior \( p \) it can not be computed explicitly. However it can be shown that the KL divergence can be minimized by maximizing the so called evidence lower bound (ELBO) denoted by \( \mathcal{L} \) shown in Equation 2.6

\[
\mathcal{L} = E_q [\log p(D, \theta)] - E_q [\log q(\theta)]
\]

Before starting the optimization, \( q \) is chosen to be of some distribution or distributions that make sense for the latent variables in question and such that the expectations are computable. The optimization then gives the variational parameters that minimize the KL divergence.

One frequently used variant of variational inference is mean field variational inference in which it is assumed that \( q \) can be factorized over all or groups of the
2.3. BAYESIAN MODELS

![Figure 2.2: Comparison of the approximated q distribution achieved from minimizing reverse and forward KL divergence. Plots from [31], where the blue contour lines correspond to the true posterior and the red contour lines correspond to the approximated q.](image)

latent variables in $\theta$ as in Equation 2.7 where $\lambda$ holds the variational parameters for $q$. Each of the separate $q_i$ distributions can be of different types and are chosen to match the distribution that the model that is being approximated posits for the latent variables.

$$q(\theta|\lambda) = \prod_{i=1}^{m} q_i(\theta_i|\lambda_i) \quad (2.7)$$

This means that possible dependencies between the latent variables defined in the model are disregarded and the approximation of the posterior will not be able to capture any correlation between latent variables. However, in many cases this does not matter very much since given enough data variance of latent variables is small which then implies that the covariance is also small [12].

Commonly the maximization of the ELBO in mean field variational inference is done through coordinate ascent which means that each variational distribution $q_i$ is maximized while the others are kept fixed. Maximizing $q_i$ is done by finding the variational parameters $\lambda_i$ that maximizes the ELBO. Thus for maximizing the ELBO for $q_i$ the objective function to maximize becomes as shown in Equation 2.8 where the $-i$ notation means everything but $i$. From the objective function for variational distribution $q_i$ the update rules for variational parameter $\lambda_i$ are derived. This coordinate ascent gives a local maximum of the ELBO function.

$$\mathcal{L}_i = \int q_i(\theta_i|\lambda_i)E_{-q_i}[\log p(\theta_i|\theta_{-i},\mathcal{D})] \, d\theta_i - \int q_i(\theta_i|\lambda_i)\log q(\theta_i|\lambda_i) \, d\theta_i \quad (2.8)$$
2.4 Graphical Models

Graphical models [4] are a graphical notation used to simplify the writing of complicated algebraic expressions of a probabilistic model. They are often used to illustrate topic models.

The two main parts of a graphical model are nodes and edges. The nodes represent random variables and the edges represent the probabilistic relationships between the variables. In this thesis, only directed acyclic graphical models, or Bayesian networks, are considered. Thus an edge $a \rightarrow b$ represents a conditional dependency where the distribution over $b$ is conditioned on $a$. In this way the graphical model easily visualizes how the joint distribution over all the random variables can be written as the product of each variable’s distribution where each such distribution only depends on a subset of the variables. To give an example, the joint distribution of the variables in the graphical model depicted in Figure 2.3 is $p(a, b, c, d) = p(c|a, b)p(d|b)p(b|a)p(a)$.

More complex models sometimes include many variables that all have a conditional dependency on one other variable. To express this kind of dependency graph more compactly, graphical models use the plate notation. This is depicted in Figure 2.4 where 2.4b and 2.4a represent the same model.

This paper uses the common convention of using shaded circles to represent observed variables and non-shaded circles to represent the latent variables.

2.5 Probabilistic Topic Models

Probabilistic topic models are a type of probabilistic model that try to discover a thematic latent structure of documents [5] that can then be used for doing orga-
2.5. PROBABILISTIC TOPIC MODELS

...mization, information retrieval, classification, gaining insights etc in large corpuses of these documents. Topic modeling was first developed in the context of natural language processing (NLP) but has later been applied to other problem domains such as computer vision [6,13,39] and computational biology [34]. Due to this origin, the structure of the data that is being modeled is in the form of a collection of documents where each document is a collection of words. While there are many different types of topic models, the latent representation that a topic model finds for a document is a distribution over topics. A topic is defined as a distribution over a fixed vocabulary.

Most topic models assume documents are simply bags of words (multisets), which implies that the order of the terms is not considered in the model. It should be noted that work has been done to account for this ordering in topic models [46].

**Latent Semantic Analysis (LSA)** [10] laid the foundation for much of the work in topic models. It makes the assumption that semantic information can be extracted from the term-document co-occurrence matrix \( X \) where each row represents a document and each column a word in the vocabulary. LSA does this by computing the singular value decomposition on this matrix, \( X = U \Sigma V^T \), where the diagonal \( \Sigma \) contains the eigenvalues of \( X X^T \) which in turn encodes the variance captured by the principal components of the term co-occurrence matrix \( XX^T \). \( U \) and \( V \) can be interpreted as containing the term-topic associations and document-topic associations respectively. A lower rank representation of a document is obtained by only considering the largest values of \( \Sigma \).

**Probabilistic Latent Semantic Analysis (PLSA)** [21] is a probabilistic version of LSA. PLSA works by defining a joint probability model over pairs of document-word observations, \( p(d, w) = p(d) \sum_z p(w|z)p(z|d) \). The relationship with standard LSA is seen by rewriting the joint probability as \( p(d, w) = \sum_z p(z)p(d|z)p(w|z) \) and noting that \( \Sigma, U, \) and \( V \) from LSA represents \( p(z), p(w|z), \) and \( p(d|z) \) respectively. A graphical model of PLSA is presented in Figure 2.5. This model represents a generative process where a topic \( z \) is drawn with probability \( p(z) \) and then the document index \( d \) and word \( w \) are drawn independent of each other but dependent on the topic \( z \).

As Blei et al. point out in [7], PLSA does not provide a probabilistic model at the document level, meaning there is no probability distribution over the topic distribution for a document. This leads to overfitting due to the number of parameters (the topic mixture parameters for each document) increasing linearly with the corpus size and no clear way of assigning probability to a document outside the training data. Solving these problems, by being more Bayesian and using a prior distribution for the topic distribution, were their motivation for proposing the Latent Dirichlet Allocation model explained next.

**Latent Dirichlet Allocation (LDA)** [7] is one of the earliest and simplest probabilistic topic models. Many later topic models with further assumptions about the data have been based on LDA [6,13,30]. It is based on the idea that a document can be represented as a mixture of the latent topics with different proportions where different topics have different probability distributions over the words in the
vocabulary. LDA is easily illustrated by its graphical model in Figure 2.6 and by its
generative process which is as follows. First at the corpus level, the word probability
distributions for each topic, $\beta_k \sim \text{Dir}(\sigma)$, are drawn from Dirichlet priors and then
for each document:

1. Draw the multinomial topic distribution, $\theta \sim \text{Dir}(\alpha)$

2. For each word position in the document
   a) Draw the topic, $z \sim \text{Mult}(\theta)$
   b) Draw the word, $w \sim \text{Mult}(\beta_z)$

### 2.6 Multimodal Data

Multimodal data is data where there exists multiple representations of some object.
These representations can be of the same or of different types and their descriptive
power of the data can differ. There may exist correlation between modalities but
they can also be independent and similarly there can also exist variation in the
modalities that is private to them.

A simple example of multimodal data is images with associated annotations.
In this case it is likely that the two modalities do correlate in some way since an
annotation is usually descriptive of what the image depicts. At the same time the
annotation might contain stopwords which may not correlate with any information
in the image and there might be aspects of the image which are not reflected in the
annotation.

In this thesis views and modalities are sometimes used interchangeably.
2.7 Image Features

The raw data of images is the intensity of each pixel (or intensities of each color channel if it is a color image). For many models, using this way of representing the data of an image quickly becomes too large to process. Instead, different methods to extract another smaller representation can be employed. The intention is then to preserve or amplify information needed for subsequent learning algorithms.

2.7.1 Handcrafted Features

*Scale Invariant Feature Transform* (SIFT) is an algorithm for extracting feature descriptors from an image. Since its development it has been one of the standard feature extraction methods for many tasks in computer vision. The SIFT descriptors are invariant to translations, rotations, and scaling transformations. It is also robust to variations in illumination. The original formulation included a method to find interest points in an image to extract the descriptors from, but SIFT can also be used to extract descriptors from a grid of points (dense SIFT).

*Patch intensity histograms* computes a histogram of the pixel intensities in a patch in an image. These histograms can be computed for patches in a sliding window manner over an image to give local representations in the image. This method is parametrized by the bin size of the histogram.

2.7.2 Deep Convolutional Neural Network Extracted Features

Deep convolutional neural networks (CNN) have proven to be a very strong model for both classification and regression tasks in multiple domains but especially in computer vision. A big advantage of CNNs is that manual feature extraction is not needed, instead the CNN is simply fed the raw data directly. CNNs then learn to find a way to represent the data which they then internally use for classification or regression. This is achieved by applying filters on the input hierarchically such that each layer finds activations based on the previous layers which then builds increasingly complex features.

Recent work has shown that these generic descriptors obtained by using the activations at some layer of a CNN can be successfully used as general feature descriptors in another model as a replacement of the handcrafted features like SIFT.

2.8 Visual Words

As previously stated, topic modeling has its origin in NLP where the structure of data is documents consisting of words. Thus, when applying topic modeling to computer vision, a preprocessing step is needed, in which images are translated to the form of a document of words. The vocabulary of visual words are computed by first choosing a way to extract feature descriptors from images (e.g. SIFT). These descriptors are then extracted from all or a representable subset of the images in
the corpus and then clustered into $V$ clusters, where $V$ represents the defined vocabulary size. The cluster centers along with the used feature extraction algorithm then form a codebook. Each cluster center in the codebook corresponds to a visual word. In order to translate a given image to a document of these visual words, the feature descriptors are extracted and then matched to their closest cluster whose index is then the index of the visual word. This gives an image the same type of representation as a text document which makes it ready for use in a topic model.

2.9 Topic Modeling for Classification

IBTM and topic modeling in general gives a lower dimensional latent representation of documents. This representation can then be used as input for a separate classification algorithm like Support Vector Machines (SVM) or K Nearest Neighbors (KNN).

It should be noted that some topic models like [13, 30] incorporate a class or response variable in which case that variable could be used for a classification or regression task. But no such variable is present in IBTM, so an external classification method must be used.
Chapter 3

Inter-Battery Topic Modeling

The Inter-battery topic model extends LDA by learning a factorized latent representation using multimodal data. In IBTM each document has two views, not necessarily of the same type, where the words of each view are the observed variables. As in LDA, it is assumed that each word is sampled from a word distribution conditioned on a topic. In this model however, the concepts of shared and private topics are used. The shared topics are used to explain variation in the two data views that is shared between them and the private topics are used to explain variation that is only present in the individual views. Each view has its own vocabulary and the private and shared topics of the first view define distributions over the first view’s vocabulary and similarly for the second view. Thus words that are common in many data instances of a certain class are likely to have a higher probability in one or more of the shared topics’ distributions and lower probability in the private topics’ distributions. From which word distribution a word is drawn from depends on whether a shared or private topic was drawn for that word. Each document has two variables, one for each view, that control the extent of topic sharing meaning it affects how often a shared or private topic is drawn for a view in the document.

A visualization of IBTM is given as a graphical model in Figure 3.1. The outer plate corresponds to the document and the two inner plates correspond to the two views of a document. The two variables inside the two inner plates are the topic assignment variable and the observed word variable, which corresponds to the inner plate of the LDA model. Thus $w$ and $a$ are the observed words from the two views and $z$ and $y$ are the respective topic assignments for those words. The private topic distributions for each view are denoted by $\kappa$ and $v$ and the shared topic distribution is denoted by $\theta$. Both the shared and private topic distributions for each document are drawn from Dirichlet distributions that are all parametrized by separate hyperparameters. The word distributions are denoted by $\zeta$, $\beta$, $\eta$, and $\tau$ where $\beta$ and $\eta$ are the word distributions for the shared topics for the two views and $\zeta$ and $\tau$ are the word distributions for the private topics. All the word distributions are also drawn from Dirichlet distributions parametrized by separate hyperparameters but at a corpus level. The two control parameters are denoted by $\rho$ and $\mu$ and are
CHAPTER 3. INTER-BATTERY TOPIC MODELING

both drawn from Beta distributions parametrized by separate hyperparameters.

The model is well-illustrated through its generative process which for IBTM consists of first drawing the word distributions for the shared and private topics of both views, \( \zeta_t \sim \text{Dir}(\sigma_{p1}) \), \( \beta_k \sim \text{Dir}(\sigma_{s1}) \), \( \eta_k \sim \text{Dir}(\sigma_{s2}) \), \( \tau_s \sim \text{Dir}(\sigma_{p2}) \), and then for each document

1. Draw the shared topic distribution, \( \theta \sim \text{Dir}(\alpha_s) \)
2. Draw the private topic distribution for the first view, \( \kappa \sim \text{Dir}(\alpha_{p1}) \)
3. Draw the private topic distribution for the second view, \( v \sim \text{Dir}(\alpha_{p2}) \)
4. Draw the partition parameter for the first view, \( \rho \sim \text{Beta}(\iota_{11}, \iota_{12}) \)
5. Draw the partition parameter for the second view, \( \mu \sim \text{Beta}(\iota_{21}, \iota_{22}) \)
6. For each word position in the first view
   a) Draw the topic, \( z \sim \text{Mult}([\rho \theta, (1 - \rho) \kappa]) \)
   b) Draw the word, \( w \sim \text{Mult}(\beta_z) \) if \( z \) is a shared topic, else \( w \sim \text{Mult}(\zeta_z) \)
7. For each word position in the second view
   a) Draw the topic, \( y \sim \text{Mult}([\mu \theta, (1 - \mu) \nu]) \)
   b) Draw the word, \( a \sim \text{Mult}(\eta_y) \) if \( y \) is a shared topic, else \( a \sim \text{Mult}(\tau_y) \)

This generative process corresponds to the joint probability distribution given in Equation 3.1 where the variables have the following meaning.

\[
\begin{align*}
T & \quad \text{The number of private topics in the first view} \\
K & \quad \text{The number of shared topics}\\
S & \quad \text{The number of private topics in the second view} \\
M & \quad \text{The number of documents} \\
N & \quad \text{The number of words in the first view} \\
L & \quad \text{The number of words in the second view} \\
\end{align*}
\]

\[
p(\kappa, \theta, \nu, \rho, \mu, \zeta, \beta, \eta, \tau, z, y, w, a) = \frac{\prod_{t=1}^{T} p(\zeta_t | \sigma_{p1}) \prod_{k=1}^{K} p(\beta_k | \sigma_{s1}) \prod_{k=1}^{K} p(\eta_k | \sigma_{s2}) \prod_{s=1}^{S} p(\tau_s | \sigma_{p2})}{\prod_{m=1}^{M} p(\theta_m | \alpha_s) p(\kappa_m | \alpha_{p1}) p(\nu_m | \alpha_{s2}) p(\rho | l_{11}, l_{12}) p(\mu | l_{21}, l_{22}) \prod_{n=1}^{N} p(z | \beta_m, \kappa_m, \rho_m) p(w | z, \zeta) \prod_{l=1}^{L} p(y | \theta_m, \nu_m) p(a | y, \eta, \tau)}
\] (3.1)

\(^1\text{Comma denotes concatenation of the two vectors}\)
3.1. INFERENCE

As stated earlier, the model makes no assumptions about the type of data in the two views since they have their own vocabularies. This makes the model flexible in the sense that it can be used to model data where the views are e.g. imagedata/annotation, imagedata/imagedata or even the same type of data but with different descriptors, etc. Chapter 5 explains how the model is used in this thesis.

The fact that the two control variables, \( \rho \) and \( \mu \), for the topic sharing are used means that each view can have different levels of shared information. Consider for example data consisting of Wikipedia articles where the two views are the text in the article together with an associated image as in [43]. In this case the data might have a weak relationship between the two views because many articles will have large amounts of text which does not directly relate to the contents of the image and being able to represent these different levels of relevance could improve performance. These control parameters range from 0 to 1, being drawn from a Beta distribution. If such a control parameter goes to 0, the view will only use its private topics and none of the shared topics and if it goes to 1 only the shared topics will be used.

3.1 Inference

As discussed in Section 2.3, to compute the posterior distribution over the latent variables for Bayesian models is often intractable. This is the case for IBTM as well and in the original work [50] approximative inference is done with mean field
variational inference. The variational distribution is thus defined as in Equation 3.2 where the variational parameters have been omitted.

\[ q(\kappa, \theta, v, \mu, \gamma, \rho, \tau, z, y) = q(\kappa)q(\theta)q(v)q(\mu)q(\gamma)q(\rho)q(\tau)q(z)q(y) \quad (3.2) \]

This is then used to form the ELBO which is then iteratively updated with coordinate ascent on the variational parameters in order to maximize it until a local optimum has been found.

Depending on the task at hand, some of the latent variables are not of immediate interest but are needed for the inference of the model. These are called nuisance parameters. For a classification or information retrieval task where the topic distribution is used to represent a document \( m \), the interesting parameter might be \( \theta_m \).

In this case where the inference is done with variational inference and \( q(z_{mn}|\phi_{mn}) \) and \( q(y_{ml}|\chi_{ml}) \) are posited as the variational distributions over the topic assignment variables of the observed words in the two views, the variational parameters \( \phi_{mn} \) and \( \chi_{ml} \) will then be used to compute the approximation of the latent variable \( \theta_m \) by \( \alpha_s + \sum_n \phi_{mn} + \sum_l \chi_{ml} \) and then normalizing it.
Chapter 4

Related work

This section gives an overview of other work that has been published in areas with problems similar to the ones that IBTM deal with. This includes work dealing with multimodal data, supervision of topic models, and factorizing solutions. Both supervised methods and factorized methods have the goal of learning a representation with further separation of classes just like IBTM. While LDA might not be used on its own to a great extent anymore, it is still a common framework that others have used when developing new topic models.

4.1 Topic Modeling

Among the supervised topic models there are in general two techniques to incorporate the supervision, upstream supervision and downstream supervision. In upstream supervision the latent topics are dependent on the label and in downstream supervision the label is dependent on the topic assignments. One of the first supervised topic models was proposed in the paper *Supervised Topic Models (SLDA)* [30]. It builds upon LDA to provide a better topic model to use for prediction of documents paired with a response. This is done by adding an observed response variable for each document to the model in a downstream manner such that the document and its response are jointly modeled. The generative process is such that topic assignments and words are first drawn in the same way as in standard LDA and then the response variable is drawn conditionally on the drawn topic assignments. This technique was found to improve prediction of the response value given a document when compared to applying linear regression on the topic distribution for a document inferred using standard unsupervised LDA. This result showed the benefits of incorporating a label or response in a dimension reduction technique when prediction is the end goal.

Another approach to supervising a topic model is the work done in *A Bayesian Hierarchical Model for Learning Natural Scene Categories* [13] which introduces two models both of which extends LDA by introducing a category variable for classification. The two models differ in their slightly different generative processes
where the first one prepends the generative process of LDA by first drawing a category label and then drawing a topic distribution from a Dirichlet conditioned on the category label. The rest of the generative process is the same as in LDA. The second model has a separate topic-word distributions for each category $c$ and thus the drawn category also influences the distribution over words, in addition to the topic assignment. The main difference between these models and SLDA \cite{slda} is that the supervision here happens upstream meaning it affects the topic distribution drawn for a document while the downstream supervision of SLDA lets the topic assignments affect the response variable.

In \textit{How to Supervise Topic Models} \cite{howtosupervise} the authors analyze the behavior and impact of supervision in topic models using SLDA \cite{slda}. It is first pointed out that the goal when using supervised topic models is to learn better representations of the data that are suitable to the task. This often means to find a representation that increases the signal to noise ratio which would better separate different classes in the latent space. In order to better be able to study the effectiveness of supervision in SLDA a variant of it is developed, Power-SLDA (PSLDA). The difference here is that the response variable of PSLDA is drawn multiple times instead of only one time as in SLDA. By varying how many times it is drawn it can be studied how much the supervision influences the topic representation. Their experiments show that SLDA and standard LDA learn similar topic representations. It is also shown that PSLDA is able to boost the supervision which gives a positive effect on the topic representation but the effect is highly dependent on the data. Data with low signal to noise ratio benefits more from the boosted supervision.

The same paper also introduces two supervised factorized topic models. Both are extensions of SLDA with slight differences to incorporate the factorization. The first one, NUFSLDA, assumes that only a subset of the topics should be shared between the observed words and the label. The other topics would then explain the noise, i.e. the words that are not relevant to the label. The other one, NCFSLDA, works like NUFSLDA but with the added constraint to the model that the noise should be structured by letting the noise respond to a noise class. These two models are then shown to be more robust than PSLDA when using a sufficient number of topics.

Exploring data consisting of multiple modalities or with more than one view is increasingly common and many topic models have been developed to handle this. One of the earliest ones was \textit{Modeling Annotated Data} \cite{modellingannotated}. It examines the problem of modeling annotated data where annotated data is described as data of two types where instances of one of the types describes instances of the other type. The paper first describes the strengths and weaknesses of two hierarchical models, a gaussian-multinomial mixture (GM-mixture) model and a gaussian-multinomial LDA (GM-LDA), where in both models, the words from the two views are considered independent given a common causal variable. This is done for three different tasks, modeling the joint distribution for an image and its annotation, modeling the conditional distribution over words in an annotation given the image, and modeling the conditional distribution over words in an annotation given a region of the image.
4.1. TOPIC MODELING

They find that the GM-LDA model models the joint probability of an image and its annotation better than the GM-mixture but that GM-mixture model performs better at generating annotation given an image. A third model is then proposed, named correspondence-LDA (corr-LDA), that combines the strengths of the two other models. This is achieved by using a generative process that first generates the regions of the image and then for each annotation word chooses one of the generated regions and generates a corresponding word conditioned on the topic assignment of the chosen region.

*Simultaneous Image Classification and Annotation* [44] combines the problems of dealing with multimodal data with that of classification. It considers image data that is labeled with both a class label and annotation. Under the assumption that in such data, the class label works as a global description of the image and the annotation words work as local descriptions of regions of the image, they propose a topic model that jointly models an image, the corresponding annotation, and its class label. The model combines ideas from supervised LDA (SLDA) [30] and corr-LDA [6]. They test their model both in classification and annotation tasks and find that it gives better results than two other bayesian hierarchical methods [8,13] in classification and slightly better results than corr-LDA in annotation. The results demonstrate that their model can give a predictive distribution for both class and annotation by finding a topic representation that explain both the image data as well as the annotation for the same image.

The need for being able to explain variation in the observed data into separate factors has been recognized since *An inter-battery method of factor analysis* [41] in 1958. This factorization can be used to isolate the variation that is relevant to the task and explain away the non relevant information. Some models also represent the relationships between factors [24]. Many factorizing models exist for both continuous data [22,24] and count-data like topic models operate on. This type of factorizing has been incorporated in topic models. One of which is *Factorized Topic Models* [48] which brings up the difficulties in inference in models where the observed data might not be representative of the variations in the underlying factors to be inferred. To illustrate this problem an example is given where they consider an animal classifier where the images are represented with SIFT descriptors on training data consisting of horses, cows, and cats. The SIFT descriptors will mostly pick up information from the fur textures and since each class can have e.g. spotted fur textures the classifier will have difficulties separating the three classes. Shape would be a better feature in order to separate the classes but this information is "hidden" in the more prominent texture information which in this case acts as structured noise. To address this problem a model is proposed that modifies LDA to exploit supervision in order to find a factorized latent representation of the observed data. The factorized representation separately models the variation in data that is private to a certain class from the variation that is shared between many classes. This leads to a representation that better captures the essence of the classes which makes separation easier. This is achieved by encouraging topics to either assume a high correlation with class or a very low correlation with class.
via a factorizing prior that partitions the topic space into a private and shared part. In this way, some topics will encode within-class variation and some will encode between-class variation. When using this model for inferring a class label given an image, only the private class information needs to be considered while the shared class information represents the structured noise. The model is compared to other LDA variants [13,30] and is found to perform better in all tasks due to the factorization.

Factorized multi-modal topic model [43] combines multimodality with factorization. It deals with analysis of multimodal data collections where the modalities may have weak correlations. To approach this, they propose a new topic model that learns dependencies between modalities in the form of shared topics as well as modeling variations within modalities with private topics. The model is flexible in the sense that no particular factorization structure is assumed but rather learnt from the data. The proposed model uses separate vocabularies and topic proportions for each modality but has another variable to explain how topic proportions of different modalities correlate.

Hierarchical Dirichlet Process [40] (HDP) is similar to LDA in that it finds a latent representation of documents in the shape of a topic mixture. The difference is that HDP is nonparametric which in this case means that the number of topics does not need to be specified beforehand. This is beneficial especially for data where the concept of what a topic might be is hard to tell. HDP differs from LDA mainly in that the Dirichlet distributions are replaced by Dirichlet Processes [14] which is the infinite dimensional equivalent.

4.2 Image Dating

Strictly related to the application in this thesis but with an entirely different approach is Analyzing Human Appearance as a Cue for Dating Images [37] and A Century of Portraits: A Visual Historical Record of American High School Yearbooks [18]. The methods proposed in these works are based on a deep convolutional neural network (CNN) which is a discriminative model compared to the generative nature of the previously presented papers in this section. As explained in Section 2.7.2, CNNs find custom hierarchical features to use for separation of the classes. In [37] they perform an experiment to better understand what the network actually learns, where they corrupt an image by setting all pixels in a patch to the mean intensity, feed it to the network and then compute the euclidean distance between the original output vector and the output vector of the corrupted image. This is done for all patches in a sliding window fashion in order to see which patches had the largest impact on the prediction and they concluded that areas around the eyes and hair have the biggest effect.
Chapter 5

Method

5.1 Data

To evaluate this model a dataset consisting of frontal facing yearbook photos from American highschools [18] is used. In this dataset there are 37921 images of men and women from the years 1905 to 2013 with varying frequencies. This is visualized in Figure 5.1 as a histogram of the number of images of the women of each year. Due to the very low number of images from the early years, it was decided to only use images from year 1930 and up. For performance reasons, the total number of images was reduced by arbitrarily only using the images of women.

The remaining 19807 images were split into a training set with 15467 images and a testing set with the remaining 4340 images. The split was done by taking 25% of the images from every year to the test set and the rest to the training set.

5.2 Document Creation

As mentioned in Section 2.8, in order to use a topic model or IBTM specifically on images, a preprocessing step is necessary where an image is represented as histogram of the words in the vocabulary. This process is described in Section 5.2.1 and 5.2.2. The idea is then to see images from the same time period as different views of that time period in order to try to find the information that can be used to

![Figure 5.1: Number of images of women per year.](image-url)
separate it from other time periods. Thus for the training documents, these image representations are paired once with themselves and once with a random other image representation from the same class in order to form the multimodal documents that are presented to IBTM. This way of creating the multimodal documents was chosen after seeing bad results when only pairing image representations with one other random image from the same class (and not with itself). This leads to a total of 30934 IBTM training documents. This process is not done for the test documents as the shared topic distribution can be inferred based on only the one view after training of the model has been done.

The vocabulary sizes are chosen somewhat arbitrarily but is on the same order as similar applications [13,50].

5.2.1 Using SIFT and Intensity Histogram Features

For the SIFT features and intensity histogram features this preprocessing is done by randomly choosing 30 images from every year, extracting the feature descriptors for each image, and then clustering the feature descriptors into 256 clusters using the K-Means algorithm which then forms the codebook. Each image, in both training and testing sets, are then translated using the codebook into a histogram of visual words by for each feature descriptor choosing the closest cluster centroid (Euclidean distance). Both SIFT features and intensity histogram features are extracted densely with a neighborhood size of 16 pixels and a step size of 8 pixels.

5.2.2 Using CNN Features

The CNN used in these experiments is of the VGG16 architecture [38] and is pre-trained on the Imagenet dataset [11]. Outputs at the second fully connected layer and outputs at the third pooling layer of this CNN model are extracted and used to create in total four different image representations (bag of words).

Inspired by [19], a representation of an image is obtained for both chosen CNN layers by extracting patches of the image at different scales and then feeding these
through the CNN to get the CNN’s representation of each patch at the given layer. Specifically each image is first rescaled to $256 \times 256$ pixels and the patch sizes are $256 \times 256$ (full image), $128 \times 128$, and $64 \times 64$. The CNN representations are extracted in a sliding window fashion with a step size of 32 by taking the activation at the given layer. Each image patch is then rescaled to the network’s input size of $224 \times 224$ before passing it to the network. Separate K-Means clustering is performed at each of three scale levels to form three codebooks with vocabulary sizes $V_1=25$, $V_2=50$, $V_3=50$ and $V_1=50$, $V_2=100$, $V_3=100$ respectively for two different variants of this feature. Features extracted at the first scale level are matched to an index in the codebook corresponding to the first scale level and similarly for the other two scale levels when translating an image patch’s raw feature representation into a word. For the second and third scale levels word indexes are translated up by adding $V_1$ or $V_1+V_2$ respectively, such that scale level 1 words $\in [0, V_1)$, scale level 2 words $\in [V_1, V_1 + V_2)$ and scale level 3 words $\in [V_1 + V_2, V_1 + V_2 + V_3)$. In total this gives four different image representations, two different layers with two different choices for the codebook sizes.

For the third pooling layer, the output from the CNN for a given image patch is of size $28 \times 28 \times 256$. The $28 \times 28$ matrix slice of this output tensor with the largest matrix elementwise sum is chosen and flattened row wise to form a 784 dimensional vector. The output from the CNN at the second fully connected layer is 4096-dimensional. In order to make computation more efficient, principal component analysis (PCA) is applied for both layers at all three scale levels to reduce the dimensionality to 500. The choice of layers is based on the intuition that more shallow layers represent more general features and deeper layers represent more specialized features.

The features extracted at the second fully connected layer are referred to as $fc2 V_1, V_2, V_3$ where $V_1$, $V_2$, $V_3$ are the sizes of the vocabularies at each scale level. The features extracted at the third pooling layer are referred to as $pool3max V_1, V_2, V_3$ where $V_1$, $V_2$, $V_3$ means the same thing as for the second fully connected layer.

For all types of CNN features, the subset of images to create the codebooks is chosen in the same way as for SIFT and intensity histograms. Euclidean distance is used for choosing the closest cluster centers.

### 5.3 Training IBTM

As explained in Section 3, IBTM can be used with different kinds of data in its two views. Since the dataset consists of only image data (apart from the year label) the two views in this experiment consist of image data from two different images from the same year. Images from the same class are paired as described in Section 5.2 and presented as a multimodal document to IBTM. The idea here is that IBTM will separate the variation in data that is shared between the two images from the variation that is private. Since the two images are from the same class the modeled
shared variation should be a better representation of the class.

5.3.1 Using Online Variational Inference

Due to a rather slow convergence of the mean field variational inference on this size of data, a variant of the inference algorithm is also developed where the algorithm is altered to instead handle data in batches in an online manner. This is achieved by following the changes done in [20] to LDA and [45] to HDP but in this case adapted to IBTM.

Specifically, this is done by subsampling the data to form a batch and then computing an approximation of the gradient of the corpus-wide variational parameters $\lambda$ and $\nu$ which are the variational parameters for the variational distribution of the shared topics’ word distributions. Implementation wise, these also include the updates for $\xi$ and $\omega$ which are the variational parameters for the views’ private topics’ word distributions. The approximation is based on the batch and then scaled according to the ratio of the size of the batch (S) and the size of the whole training corpus (D). The iterative updates of these parameters are given in Equation 5.1. In Equation 5.1 and 5.3 $t$ increases with every batch handled. For implementation details, multiple complete passes of the whole corpus are made but only one batch is considered when updating the corpus wide parameters. In each data pass the order of the documents is randomized and then split into the batches.

\[
\begin{align*}
\lambda &\leftarrow \lambda + \omega_t \partial \lambda(D_{batch}) \\
\nu &\leftarrow \nu + \omega_t \partial \nu(D_{batch})
\end{align*}
\]

(5.1)
5.4. CLASSIFICATION

\[ \partial \lambda_{kv}(D_{\text{batch}}) = -\lambda_{kv} + \sigma_{s1} + \frac{D}{S} \sum_{m \in D_{\text{batch}}} \sum_{n} I(w_{mn} = v) \phi_{mnk} \]
\[ \partial v_{kw}(D_{\text{batch}}) = -v_{kw} + \sigma_{s2} + \frac{D}{S} \sum_{m \in D_{\text{batch}}} \sum_{n} I(a_{mn} = w) \chi_{mnk} \]  \hspace{1cm} (5.2)

\[ \omega_t = (1 + t)^{-0.6} \]  \hspace{1cm} (5.3)

This variant of IBTM will be referred to as **Minibatch-IBTM**.

5.3.2 Merged Topic Word Distributions

Due to the fact that in this specific problem, the two views are of the same data type and thus share the same vocabularies, IBTM should statistically find the same distribution for shared topic word distributions, \( \beta \) and \( \eta \), and the same distribution for private topic word distributions, \( \zeta \) and \( \tau \). That is, \( \beta_k \) should be similar to \( \eta_k \) for example. This could also be enforced by introducing a variant of the model where the two shared topic word distributions are merged into one, and likewise for the private topic word distributions. The only changes in update equations would be to have the two views use the same variational parameters for the merged topic word distributions instead of the ones previously specific to each view.

This variant of IBTM will be referred to as **Merged-IBTM**.

5.4 Classification

The latent representation of the documents that IBTM finds are the topic distributions for the shared topics and the private topics for the two modalities. Since the private topics in this experiment are used to explain variation that is not defining for a class (year/decade) they are ignored. The shared topic representations \( \theta_m \) are represented by normalizing \( \hat{\theta} \) shown in Equation 5.4 where \( \phi_{mn} \) and \( \chi_{mn} \), as explained in Section 3.1, are the variational parameters of the variational distributions over topic assignments and \( \alpha_s \) is the hyperparameter of the prior of shared topic distributions.

\[ \hat{\theta}_m = \alpha_s + \sum_n \phi_{mn} + \chi_{mn} \]  \hspace{1cm} (5.4)

\( \theta \) is then used to train a K Nearest Neighbor classifier \[33\]. Since there are different number of documents for each class, the data points representing the documents are given a weight in the KNN classifier such that all classes are equally represented.

27
Chapter 6

Experiments

This classification with IBTM and KNN is tested for the image dating task both yearwise and decadewise under different settings. Yearwise classification in this case leads to an 84-way classification and decadewise to a 9-way classification.

The settings of the hyperparameters in the experiments are, unless stated otherwise, as follows. $\alpha_s = \alpha_{p1} = \alpha_{p2} = 0.8$, $\sigma_s_1 = \sigma_s_1 = \sigma_s_1 = \sigma_s_1 = 0.6$, $\nu_1 = \nu_2 = \nu_2 = \nu_2 = 2$. In all experiments using the non-modified version of IBTM the training is run until convergence of the total log probability of the model to a tolerance of $1e-7$ or up to a maximum of 300 iterations. For the classification performance results, for each setting the best result of 3 runs with different seeds for the random number generator is presented.

In each experiment, SIFT 256 are the visual words extracted with the SIFT algorithm with a vocabulary size of 256, Intensity 256 is the same but with the intensity histogram algorithm, pool3max 25,50,50 / 50,100,100 are the visual words extracted from the third pooling layer of the VGG16 convolutional neural network with vocabulary sizes of 25,50,50 and 50,100,100 respectively (explained in Section 5.2). fc2 25,50,50 / 50,100,100 is the same but extracted from the second fully connected layer of the same network.

6.1 Classification performance

Table 6.1 shows the classification accuracies as well as the median of the absolute differences between predicted and actual year. Table 6.2 shows the same for decadewise classification. The number of neighbors K in the KNN classifier was set to 30 and was found via K-fold cross validation on the training set for the fc2 25,50,50 feature type. The same K was used for the remainder of experiments.

The first observation that can be made from these results is that the SIFT and pool3max features seem to carry the least of the information needed to make accurate classification in the yearwise classification task. This lack of information propagates to IBTM giving substandard performance. For supervised CNNs in general, the activations of the earlier layers resembles basic edges or shapes 47.
Thus the outputs of image patches as described in Section 5.2.2 are likely similar for all images because, slightly simplified, all the images contain the same basic shapes. This in turn means that the visual words yielded from these activations will be similar for all the documents meaning there is less information to use for separating classes. The SIFT algorithm is based around histograms of gradients in the intensities of the image which could suffer from similar problems. SIFT is also invariant to illumination which could be of importance for this classification task since the quality of cameras in terms of light sensitivity have likely improved over the years.

The intensity based features show a higher accuracy than SIFT but is also more often far off in prediction. The second type of CNN features gives the best results in accuracy by some margin but also a much lower median prediction difference. For fc2 25,50,50 and fc2 50,100,100 which are created with the same feature type but with different vocabulary sizes, the finer quantization of fc2 50,100,100 seems to be beneficial.

For the intensity and pool3max features, increasing the number of topics beyond K=30 (shared), T=S=10 (private) showed no improvement whereas for the other type of features this limit seems to have been around K=50, T=S=20. There is a slight decline in performance for the highest topic settings for the fc2 50,100,100 vocabulary. This could be either because of higher risk of poor local optima due to the higher complexity of the model or that having too many topics is overfitting to the training data which could be because in that case the number of topics approaches the whole vocabulary size.

In the decadewise classification some of the relative performances change. The fc2 features are still the best but SIFT 256 sees a big improvement and gives performance close to the fc2 features. In this case it seems K=30, T=S=10 is enough for the most part. It is also interesting how the improvement is not greater than about twice that of yearwise classification.

Figure 6.1 shows a visualization of the IBTM topic representation of the test
6.2. WHAT DID IBTM LEARN?

Figure 6.1: t-SNE visualization of the topic representation of test documents using the fc2 50,100,100 vocabulary with topic settings K=50, T=S=20. Documents when using the fc2 50,100,100 vocabulary and topic settings K=50, T=S=20 in decadewise classification. The topic representations are 50-dimensional but have been visualized in 2d using t-SNE [42]. The separation of classes is visible to some extent but there is also large overlaps.

Figure 6.2 shows the confusion matrices for fc2 50,100,100 (best feature) and for pool3max 25,50,50 (worst feature) in each column respectively. Perfect classification gives a confusion matrix where only the diagonal is highlighted. This is likely not possible in this problem but a better classifier for this problem would preferably, when it is misclassifying, at least predict a class close to the actual class. A confusion matrix for a classifier with this property would display a wider diagonal instead. The actual confusion matrices for the fc2 50,100,100 vocabulary show this to a small extent but there are also predictions that are far off which is not desirable. It is however possible that a person simply looks out of her time by perhaps having an old-fashioned hairstyle which could explain some far off predictions. For the confusion matrices for the pool3max 25,50,50 vocabulary, no improvement is shown over varying settings of number of topics and predictions are very scattered.

Figure 6.3a shows the median differences of predicted year and actual year for each year and Figure 6.3b shows the classification accuracy per year. These measures give a better view of how the system performs as a whole. There is a fairly large variance both in prediction accuracy as well as in the median prediction differences over the years which could be due to there being less images from some years when training the IBTM.
Figure 6.2: Confusion matrices for yearwise classification with fc2 50,100,100 (left) and pool3max 25,50,50 (right). Classification accuracies given in respective caption.
6.2. WHAT DID IBTM LEARN?

Figure 6.3: Median differences in predictions vs actual for each year (a) and per year accuracy (b)

Figure 6.4: Plots of probability of each shared topic over the years for an IBTM representation with the fc2 50,100,100 vocabulary
6.2 What did IBTM Learn?

Figure 6.4 shows a plot of each shared topic against the year labels with the fc2 50,100,100 vocabulary with topic numbers K=50, T=S=20. For each topic and each label, the probability is averaged over all test documents’ representations. For the topics that demonstrate little change in probability over time it is likely that these topics give higher probability to visual words that are common at all time periods. The topics that do show change over time are more likely to correspond to word distributions where visual words that actually do differ between different time periods are more common. Figure 6.5 shows the same type of plot but for the topics found with LDA. Note that some topic k in IBTM can not be compared directly to topic k in LDA since there is no order of topics in any of the models. There does however seem to be topics that could be each other’s counterparts like topic 47 in IBTM and topic 37 in LDA and topic 20 in IBTM and topic 26 in LDA. This could suggest that the models have learned very similar things.

Figure 6.6 shows a plot of the word distributions of each shared topic inferred with the fc2 50,100,100 vocabulary and topic settings K=50, T=S=20. The plotted distributions are the average of the inferred shared per topic word distributions of the two views. Most of these topics give any notable probability to just a few words and some even just for one word. This is less than optimal since it means that each topic will just function as a proxy for a word (or a couple of words). Basically it means that if a document is inferred to be e.g. 20% topic 4 it just means that word 177, which has almost all probability in topic 4, makes for 20% of the words in that document. This leads to the suspicion that passing the word histogram representation of documents directly to the classifier and thus bypassing
6.2. WHAT DID IBTM LEARN?

IBTM would give similar performance in accuracy. This was also tested and shown to give an average accuracy of 19.3% with median difference of 7y compared to the 20.9% and 6.5y achieved with IBTM.

Also from Figure 6.6 it can be noted that among the scale level 1 words (the whole image) with the fc2 50,100,100 vocabulary, only word 2 and 42 were given an substantial probability in any topic (topic 0 and 15 respectively). It should be noted that the level 1 words will naturally have less probability since there is only one level 1 word per document with the way this vocabulary is constructed. To get a feel for what one of these words look like, Figure 6.7 shows images where the scale level 1 word turned out to be the same word (2). These examples all look very different and are from different time periods. It is difficult to draw any conclusions on what type of features in the pictures are important to indicate a time period based on these examples.

Looking at a correctly classified image example shown in Figure 6.8, along with the visual words of it, the five topics with the highest probabilities are 33, 39, 30, 44, 21
CHAPTER 6. EXPERIMENTS

Figure 6.8: Examples of visual words from the three scale levels for a correctly classified example. As explained in 5.2.2, words are extracted in a sliding window manner which gives the overlap shown in the figure. The word index of each patch is given in the top left corner of the patch.

Figure 6.9: Examples of word 157 with fc2 50,100,100 vocabulary

Figure 6.10: Examples of word 209 with fc2 50,100,100 vocabulary

Figure 6.11: Examples of word 72 with fc2 50,100,100 vocabulary
and for these topics the words with the highest probabilities of their respective word distributions are given in Table 6.3 together with their probabilities. In Figure 6.8c which shows the scale level 3 words, word 157 seems to correspond to patches containing eyes and finding further image patches in other random images that also turned out as word 157 confirms this as shown in Figure 6.9. As Figure 6.7 also showed, these examples also seem very different which could be an indication of a problem with the vocabulary creation. Since it seems that many different images end up in the same cluster it could either mean that (1) K-Means gives a poor clustering because the images used to compute the clusters are not representative enough, (2) too few clusters (vocabulary size) are being used meaning that the discretization of the feature space is too coarse, (3) PCA-projected activations from the CNN are too similar, or that (4) K-Means is simply not the best way to cluster the outputs from the CNN. Unfortunately no testing of these theories has been carried out.

Table 6.3: Top five topics for a correctly classified example (Figure 6.8) and the top words of those topics.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>33</td>
<td>209 (0.0347) 200 (0.0376) 9 (0.0632) 217 (0.0033) 234 (0.0012)</td>
</tr>
<tr>
<td>39</td>
<td>72 (0.9910) 4 (0.0044) 209 (0.0002) 285 (0.0001) 158 (0.0001)</td>
</tr>
<tr>
<td>40</td>
<td>121 (0.9966) 209 (&lt; 0.0001) 228 (&lt; 0.0001) 223 (&lt; 0.0001) 158 (&lt; 0.0001)</td>
</tr>
<tr>
<td>44</td>
<td>209 (0.9974) 4 (&lt; 0.0001) 228 (&lt; 0.0001) 223 (&lt; 0.0001) 158 (&lt; 0.0001)</td>
</tr>
<tr>
<td>21</td>
<td>209 (0.5955) 121 (0.0006) 228 (0.0004) 223 (0.0001) 121 (0.0001)</td>
</tr>
</tbody>
</table>

Furthermore, Table 6.3 also shows high likelihood for words 209, 121, and 72. Examples of these words are shown in Figure 6.10 and Figure 6.11. Word 72 seems to at least correspond to hairstyles, but the patches in Figure 6.11f and 6.11g look different from the others which again shows what could potentially be a problem with the vocabulary creation.

### 6.3 Comparison with LDA

Since IBTM to a certain extent uses LDA as a framework, a comparison between them shows the value of using this more complex model over the simpler LDA for this type of problem. This implementation of LDA uses mean field variational inference just as the implementation of IBTM.

The choice of number of topics to use in both models for a fair comparison is not completely straightforward. One way is to have the number of topics in LDA equal to the total number of topics in IBTM, both the shared and private. Another way is to have the number of topics in LDA equal just the shared topics of IBTM. Because of the way IBTM is used in this task, where only the shared topics are used for classification, it was chosen to use the latter way. The reason being that in this case the private topics are not used to represent a class but instead just function as nuisance parameters to explain away variation in data that is not of interest. Tables 6.4 and 6.5 show the results of this comparison for yearwise and
decadewise classification where the number of topics K in each column was the same for both IBTM and LDA. Private topic numbers T and S are not used for LDA. Features fc2 50,100,100 are used in all experiments in this section due to it giving the best results in the previous experiments. The common hyperparameters are set the same.

The results of this experiment show that the KNN classification yields slightly better performance with IBTM representations than LDA representations. This indicates that the factorizing property of IBTM successfully explains away some of the information that is not needed for classification which was the motivation for the development of IBTM.

Looking at the inferred values for the variational parameters of the Beta distributions over topic sharing control values in IBTM, $\rho_m$, of the test documents the average is about $Beta(73, 5)$. Note that for the test documents the two views are the same and thus $\mu_m$ will be very similar (statistically identical) to $\rho_m$. Samples from such a Beta distribution are skewed to values closer to 1 ($E[\rho] = \frac{73}{73+5} \approx 0.935$). This means that for most documents most probability is given to the shared topics, which in turn means that the private topics are not used to a great extent. This could explain why results are similar to LDA (but still slightly better). One reason for the parameters to look like this is likely due to the way the pairing of the images was done to create the training documents.

### 6.4 Hyperparameters Experimentation

The fc2 50,100,100 vocabulary is used in this experiment as well. The number of topics are set to K=50, T=S=20 for all experiments. Table 6.6 shows yearwise classification accuracies and median difference for different settings of $\alpha_s$ and $\sigma_s$ which are the hyperparameters for the priors of the shared topic distribution and shared per topic word distributions respectively. The table cells with a dash corresponds to settings that were not tested.

Changing $\alpha_s$ and $\sigma_s$ seems to have made only small differences in classification performance. It is slightly more sensitive to lower values of $\alpha_s$. Asuncion in $^2$ showed that lower values of $\alpha$ and $\sigma$ in LDA when doing inference with mean field variational inference often yields suboptimal performance which is likely an explanation of these results. $\alpha_s = 1.6$ seems to be too big since this favors shared topic distributions that are not so sparse making the document representations more...
6.5. PERFORMANCE WITH MINIBATCH-IBTM

<table>
<thead>
<tr>
<th>$\alpha_s = 0.3$</th>
<th>$\alpha_s = 0.55$</th>
<th>$\alpha_s = 0.8$</th>
<th>$\alpha_s = 1.6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho_s = 0.3$</td>
<td>19.6% 7y</td>
<td>20.2% 7y</td>
<td>20.2% 7y</td>
</tr>
<tr>
<td>$\rho_s = 0.45$</td>
<td>19.7% 7y</td>
<td>20.4% 7y</td>
<td>20.9% 6.5y</td>
</tr>
<tr>
<td>$\rho_s = 0.6$</td>
<td>19.9% 7y</td>
<td>20.7% 7y</td>
<td>20.9% 6.5y</td>
</tr>
<tr>
<td>$\rho_s = 1.2$</td>
<td>-</td>
<td>-</td>
<td>21.0% 6.5y</td>
</tr>
</tbody>
</table>

Table 6.6: Yearwise classification accuracies of IBTM with different settings of $\alpha_s$ and $\sigma_s$. Similar to each other. A higher $\sigma_s$ did little difference.

The $\iota$ hyperparameters are parameters for the Beta priors which influences values of $\rho$ and $\mu$ which controls the level of sharing of topics for a document. A low $\rho$ means less probability of the shared topics and vice versa. Table 6.7 shows the results of two different settings of these parameters compared to the default values used in the other experiments. While few experiments of these parameters were performed, the settings are such that two extremes of this prior were tested, i.e. one prior that is skewed to values closer to 0 and one for values closer to 1. A very slight improvement in both accuracy and median prediction difference is observed for the values of $\iota$ hyperparameters that are skewed towards less probability on the private topics. This is to be expected since Section 6.3 showed that most probability was placed on the shared topics when using the less informative prior Beta(2, 2), thus a model with a prior that better fits the data will have higher likelihood. The opposite is true for the other setting which then gives a model that fits the data worse and thus get a worse performance.

6.5 Performance with Minibatch-IBTM

Like for the hyperparameter experimentations, only the fc2 50,100,100 vocabulary is used for the Minibatch-IBTM performance experimentation. Table 6.8 and Table 6.9 show yearwise and decadewise classification accuracies. The Minibatch-IBTM is tested with two different batch sizes, 500 and 2000. Both are run for 6 passes over the corpus. The performance is unsurprisingly better with the larger batch size but in neither case does the Minibatch-IBTM reach the same performance as the normal IBTM. For both the 500-batchsize version and the 2000-batchsize version the best results are with the K=30, T=S=10 topic settings. Perhaps this is due to there being enough topics but less total parameters for that setting which makes it less susceptible to go in the wrong direction when updating the corpuswide parameters with respect to just one batch.

Figure 6.12 shows the convergence of the log likelihood for both the normal IBTM and the Minibatch-IBTM (both batch sizes). Both versions of Minibatch-
CHAPTER 6. EXPERIMENTS

Figure 6.12: Convergence of log likelihood for Minibatch-IBTM (batchsize given in following parenthesis) and normal IBTM. (b) is zoomed in.

Table 6.8: Yearwise classification performance comparison of Minibatch-IBTM and IBTM representations.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>9.6% 11y</td>
<td>8.6% 12y</td>
<td>8.6% 12y</td>
</tr>
<tr>
<td>30</td>
<td>12.7% 10y</td>
<td>16.8% 8y</td>
<td>16.9% 8y</td>
</tr>
<tr>
<td>50</td>
<td>10.7% 11y</td>
<td>17.0% 8y</td>
<td>20.9% 6.5y</td>
</tr>
<tr>
<td>100</td>
<td>7.6% 12y</td>
<td>14.9% 8y</td>
<td>20.2% 6.5y</td>
</tr>
</tbody>
</table>

Table 6.9: Decadewise classification performance comparison of Minibatch-IBTM and IBTM representations.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>34.3%</td>
<td>34.8%</td>
<td>34.2%</td>
</tr>
<tr>
<td>30</td>
<td>37.9%</td>
<td>41.9%</td>
<td>42.9%</td>
</tr>
<tr>
<td>50</td>
<td>36.0%</td>
<td>41.6%</td>
<td>44.0%</td>
</tr>
<tr>
<td>100</td>
<td>34.8%</td>
<td>38.7%</td>
<td>34.5%</td>
</tr>
</tbody>
</table>

IBTM show fluctuation in likelihood between iterations since each update of the corpuswide parameters are based on just the batch subset of documents. The 500 batch size version does not see any improvement after just a few iterations while the larger batch size increases its likelihood overall and looks like it is still increasing at the time it was stopped in this experiment.
6.6 Performance with Merged-IBTM

In this experiment the variant of IBTM with merged topic word distributions, described in Section 5.3.2 is used with the fc2 50,100,100 vocabulary and otherwise default parameter settings. The results in Table 6.10 show similar results to the normal IBTM which suggests that the per topic word distributions for the two views in standard IBTM was already reaching similar numbers which made the change unnecessary in this case.

Table 6.10: Yearwise classification accuracies of Merged-IBTM and IBTM side to side

<table>
<thead>
<tr>
<th>K = 10, T=S=5</th>
<th>Merged-IBTM</th>
<th>IBTM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>9.2%</td>
<td>8.6%</td>
</tr>
<tr>
<td></td>
<td>12y</td>
<td>12y</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>K = 30, T=S=10</th>
<th>Merged-IBTM</th>
<th>IBTM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>16.7%</td>
<td>16.9%</td>
</tr>
<tr>
<td></td>
<td>8y</td>
<td>8y</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>K = 50, T=S=20</th>
<th>Merged-IBTM</th>
<th>IBTM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>20.9%</td>
<td>20.9%</td>
</tr>
<tr>
<td></td>
<td>6.5y</td>
<td>6.5y</td>
</tr>
</tbody>
</table>
Chapter 7

Summary and Conclusions

In this thesis, the Inter-Battery Topic Model has been applied and evaluated to a difficult classification problem with high intra-class variation and low inter-class variation. The overall accuracy with the best setting of hyperparameters and features is close to 21% for years and close to 45% for decades, however there is also a fairly large uncertainty in the predictions as they can be far off and the predictions are not equally good for all labels.

7.1 Discussion

Due to the nature of this classification task with its subtle or in some cases perhaps nonexistent information needed to accurately predict the year or decade it is difficult to know what a reasonable performance is. In a small-scale (two) test of non-expert human performance in the same tasks, a best result of 7.1% accuracy (median difference 8y) and 31% decadewise accuracy was achieved. A person trained on, say decade defining hair styles, might do considerably better in this task. [18] reached 11.3% accuracy and 4y median prediction difference from ground truth year in yearwise classification using a CNN. However, these results are not directly comparable due to differences in how training and testing sets were created. The choice of reporting the median difference in prediction from the ground truth year was chosen because both because intuitively it makes sense that a prediction close to the actual year is better than a prediction far off and also because it was used in [18]. This performance measure could however be problematic in the case of sharp discontinuities in fashion or in the case of cycles in fashion of hairstyles for example.

The experiment in Section 6.3 showed that IBTM performed better than LDA in this task, which is a positive result since IBTM is an extension of LDA with the aim of better separating the relevant from the non relevant information. These improvement are however very slight and experiments in Section 6.3 and Figures 6.4 and 6.5 showed that the two models learned similar things. There could be a number of reasons for why a larger improvement was not observed. First the image features
used might not have been entirely adequate for this task since as Section 6.2 showed one visual word could represent many perceptually different image patches. Secondly, IBTM is a much more complex model, both in that the larger number of variables gives a higher degree of freedom but also in that the model contains more dependencies between these variables and this could have implications on how well the inference algorithm works. The mean-field approximation generally is a stronger assumption for a more complex model which could lead to finding solutions that are further from the actual posterior distribution of the sought latent variables. Local optima could potentially also be problematic but the multiple differently seeded runs was an attempt to defend against this.

Regarding the feature extraction and vocabulary creation, there are mainly two ways in which the potential problems with it could arise (as discussed briefly in Section 6.2). Either it stems from the fact that the activations from the CNN for perceptually different image patches are too similar which is a reasonable suspicion considering that the used pretrained CNN was trained on an image dataset consisting of 1000 different classes where only one or a few would represent something like 'face' or 'human'. Since the CNN is trained on different examples of these classes its internal representation of them at the layer used for the vocabulary creation could likely be similar. [1] is another CNN architecture used for facial recognition which would likely work better in this task. It is however not as clear if the same sliding window technique for creating the visual words could be applied using the pretrained version of that CNN since it has been trained on whole faces. The other possible issue is with the quantization to form the words. This was done with K-Means which could give a bad clustering/quantization result depending on how the data looks. For example, if data that should semantically be assigned to the same cluster does not have spherical variance K-Means will likely split that cluster into more than one because K-Means tries to minimize the sum of squared distances to a points assigned cluster centroid. A different type of clustering is hierarchical clustering which would work better for data which is not spherically shaped which is something that should be tested. The "no free lunch"-theorem likely applies here however.

The fact that the data was unbalanced in number of instances per label most likely had implications on the performance. Figure 6.3 showed that the performance in terms of both accuracy as well as prediction difference was not consistent over the labels.

Furthermore, the possibilities of the system picking up on artifacts in the images that are telling of time periods must be considered. The raw dataset has been created by first cropping and then aligning the images according to facial landmarks in order to create a standard shape and facial pose [18]. This can possibly introduce artifacts because if a certain high school in a certain time period would have taken their yearbook photos in a specific way that would require a transformation that would e.g. introduce some kind of pattern in the right corner of the images, then those patterns might erroneously help the classification and give a misleading accuracy. In order to make a fair comparison to the results of [18] the split of the
7.2. FUTURE WORK

Data into a training and testing set should be made with these concerns in mind.

The online variant of the inference for IBTM (Minibatch-IBTM) did not reach the same performance as the standard IBTM, and seemed to be sensitive to batch size settings. This sensitivity is most likely due to the nature of the data but the implementation should be tested on another dataset to show the validity of the inference and to see that the implementation is bugfree. There was not enough time to do this in this work however.

The variant of IBTM with merged topic word distributions for the two views reached about the same performance of normal IBTM. Theoretically the idea is still good and should be an option in any implementation of IBTM as it reduces the complexity of the model at no cost of expressiveness for problems where the data views share the same vocabulary. The Minibatch-IBTM would likely benefit more from this change when it is applicable because of its more approximative method.

The way the pairing of the images was done is something that can be experimented more with as well and can have a big effect on what IBTM learns. Initially, pairing of images within a class (year/decade) was done by just permuting the training data within the class, i.e. the images were not paired with themselves and only to other images in the class. This did not get any good results since very little information (with the used features) seems to have been shared between photos from the same year (the high intra-class variation). This instead led to the type of pairing described in the method chapter which gives a topic representation similar to LDA but with slightly better extraction of the relevant information. For other problems (where there is also only one actual modality) it is probably better to not pair data instances with themselves since this would better be able to find the relevant information.

7.2 Future Work

As for future work, firstly most of the experiments performed in this thesis could probably benefit from being performed again where the concerns regarding the vocabulary creation has been addressed. Further experiments where the model is applied in other types of problems and used in different ways should be investigated. Although it is not presented in this thesis (as the results were not finished), the model was also tested on a dataset consisting of news articles about different meta-topics in the Israeli-Palestinian conflict where each meta-topic is written about from both sides. Each pair of articles describing some meta-topic was then used as the two views. By using IBTM in this way, the per private topic word distributions could be examined as a way to explore the different perspectives in the conflict.

For future work on the model itself, more advanced or robust inference algorithms could be explored in order to find out if the posterior is currently being underestimated (for this task), like structured variational inference [17]. To increase the model’s flexibility it could be extended to allow for an arbitrary number of data views. A nonparametric version of IBTM could be developed to alleviate
the process of choosing the number of topics which in the case of an arbitrary number of views could be a big benefit but also in general for data where the concept of a topic is hard to define or semantically interpret. Hyperparameters could be optimized over instead of fixed. This could allow for using a non-symmetric prior for the Dirichlet distributions which in the case of the prior over the topic distributions for documents could mean that some topics could go towards zero probability effectively turning them off which would mimic the effects (if enough topics to start with) of HDP but in a simpler model.
Bibliography


[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Im-
agenet: A large-scale hierarchical image database. In Computer Vision and 


[13] Li Fei-Fei and Pietro Perona. A bayesian hierarchical model for learning natural 
scene categories. In Computer Vision and Pattern Recognition, 2005. CVPR 
2005.


[15] Bela A Frigyik, Amol Kapila, and Maya R Gupta. Introduction to the dirich-
let distribution and related processes. Department of Electrical Engineering, 
University of Washington, UWEETR-2010-0006, 2010.


[18] Shiry Ginosar, Kate Rakelly, Sarah Sachs, Brian Yin, and Alexei Efros. A 
century of portraits: A visual historical record of american high school year-
books. In Proceedings of the IEEE International Conference on Computer 
Vision Workshops, pages 1–7, 2015.

[19] Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik. Multi-scale or-
derless pooling of deep convolutional activation features. In Computer Vision– 

[20] Matthew Hoffman, Francis R Bach, and David M Blei. Online learning for la-
tent dirichlet allocation. In advances in neural information processing systems, 

Fifteenth conference on Uncertainty in artificial intelligence, pages 289–296. 

with structured sparsity. In Advances in Neural Information Processing Sys-
BIBLIOGRAPHY


