Count Data Modeling and Classification Using Statistical Hierarchical Approaches and Multi-topic Models

Ali Shojaee Bakhtiari

Presented

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical and Computer Engineering

at

CONCORDIA UNIVERSITY

April 2014

CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Ali Shojaee Bakhtiari

Entitled: Count Data Modeling and Classification Using Statistical Hierarchical Approaches and Multi-topic Models

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Electrical and Computer Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Chair
Dr. L. Wang

External Examiner
Dr. H. Krim

External to Program
Dr. L. Kadem

Examiner
Dr. A. Ben Hamza

Examiner
Dr. D. Qiu

Thesis Supervisor
Dr. N. Bouguila

Approved by: _____
Dr. A.R. Sebak, Graduate Program Director

April 7, 2014

Dr. C. Trueman, Interim Dean
Faculty of Engineering & Computer Science

Count Data Modeling and Classification Using Statistical Hierarchical Approaches and Multi-topic Models

by

Ali Shojaee Bakhtiari

Abstract

In this thesis, we propose and develop various statistical models to enhance and improve the efficiency of statistical modeling of count data in various applications. The major emphasis of the work is focused on developing hierarchical models. Various schemes of hierarchical structures are thus developed and analyzed in this work ranging from purely static hierarchies to dynamic models. The second part of the work concerns itself with the development of multitopic statistical models. It has been shown that these models provide more realistic modeling characteristics in comparison to mono topic models. We proceed with developing several multitopic models and we analyze their performance against benchmark models. We show that our proposed models in the majority of instances improve the modeling efficiency in comparison to some benchmark models, without drastically increasing the computational demands. In the last part of the work, we extend our proposed multitopic models to include online learning capability and again we show the relative superiority of our models in comparison to the benchmark models. Various real world applications such as object recognition, scene classification, text classification and action recognition, are used for analyzing the strengths and weaknesses of our proposed models.

Thesis Supervisor: Nizar Bouguila
Title: Associate Professor

Acknowledgments

Above all I would like to thank my supervisor Dr. Bouguila for offering me the opportunity of pursuing this doctoral thesis under his supervision. I would also like to thank him for his support throughout the years of this program. His insightfulness and his kin interest in this thesis was a constant contribution towards its advancement. Above all I would like to highlight his eloquent way of dealing with research problems, which in my regard mostly contributed towards the successful completion of this thesis.

Also I would like to thank Natural Sciences and Engineering Research Council of Canada (NSERC) for partially funding this thesis.

Finally I would like to thank my great parents, my lovely sister, my dear aunt and uncle for their ever present support and my dear friends for their unshaking friendship.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The emergence of internet has led to an increasingly interconnected world. Everyday enormous amount of digital data is added to the internet data pool. The introduction of less-expensive imaging sensors inside digital cameras and cell phones has led to ever growing size of collected image databases. Facebook claims to be receiving seven petabytes of new photo content every month. As of 2012, they claimed to have stored more than 220 billion images in their servers. The same analogy can be said about huge content of textual data inside the world wide web and countless streams of digital video data generated by cheap webcams and the subsequent information flowing in the internet. While dealing with this amount of digital data, one assumes that any manual processing of these data has to be forfeited. Dealing with simple tasks one could consider machines that have a mechanical repetitive character. However, when dealing with data generated, for instance, from images captured from various view points, suffering from occlusion or affected by clutter and noise one can no longer rely on mechanical algorithms to offer reliable solutions. Learning machines and artificial intelligence on the other hand provide us with a solution for the problems mentioned thus far. One of the recurring problems in machine learning context is proper data classification. Data classification is defined as the process of assigning data to a set of predefined or evolving classes. Usually classification is performed with certain amount of uncertainty. Therefore, it is necessary that we first consider a proper measure for uncertainty. We use probability theory for describing this uncertainty.

1.1 Probability as a measure of uncertainty in data classification

Data classification in essence is a true or false problem (i.e. Certain data fit in a class or not) . What matters however is the uncertainty degree of the decision. To derive this measure one would consider probability theory [26].
For a discrete random variable (RV) X probability $P(X = x)$ defines the degree of certainty or the probability that the RV takes the value x. $P(x)$ is defined as a probability function subject to the following two conditions

$$P(x) > 0 \tag{1.1}$$

$$\sum_{x} P(x) = 1 \tag{1.2}$$

One defines conditional probability of X with respect to Y as the probability that $X = x$ provided that $Y = y$. One also defines the joint probability of X and Y as $P(x, y)$. The relationship between the conditional and the joint probability is given as:

$$P(x|y)P(y) = P(x, y) = P(y|x)P(x) \tag{1.3}$$

From the above equation we derive the Bayes equation [9]

$$P(y|x) = \frac{P(x|y)p(y)}{p(x)} \tag{1.4}$$

Bayes theorem allows us to consider priors $p(y)$ that update themselves based on the observation $X = x$. As it can be seen from the above equation. One can consider the posterior belief $p(y|x)$ as the updated version of the prior belief $p(y)$ based on the observed data $X = x$. Another concept that needs to be verified beforehand is the conditional independence. Assuming that we have a joint distribution of variables X, Y and Z depicted as $P(x, y, z)$. The variable Y is said to be conditionally

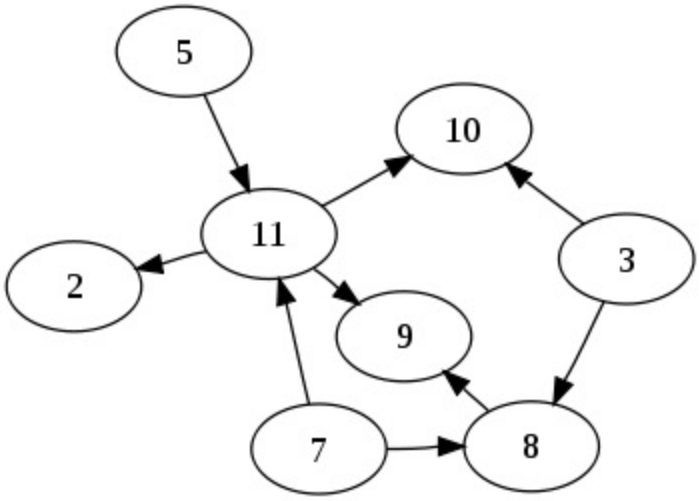


Figure 1-1: Example of a directed acyclic graph.

independent of Z given X if we have:

$$P(y|x,z) = P(y|x) \tag{1.5}$$

The concept of conditional independence leads to the introduction of graphical models. That in return are used extensively in statistical modeling.

1.2 Graphical Models and Bayesian Networks

A graphical model consists of a set of nodes and edges defining the model graph. In a probabilistic graphical model the nodes represent the variables and the edges represent the relationship between the variables.

Bayesian networks are a special form of graphical models in which the structure is represented by a directed acyclic graph (DAG). DAGs are graphs which have directed edges between nodes and have no cycles along the directed paths. An example of a DAG network is depicted in figure 1-1. Bayesian networks make conditional independence assumption. The probabilistic character of a node is decided solely by the state of the nodes directly connected to it regardless of the state of all the other nodes in the graph. Defining $Pa_i = \{Nodes|Nodes\ leading\ to\ i-th\ node\}$ and considering

the definition of conditional independence it can be shown that the joint probability distribution of the graphical model is derived as:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|Pa_i) \tag{1.6}$$

Graphical models mostly deal with special probability distributions called the exponential distributions [72]. Many common probability distributions, such as Gaussian, binomial, multinomial, Dirichlet etc. belong to this family. Assuming that $\vec{Y}$ shows the entire nodes leading to $\vec{X}$ the exponential family of distributions take the following form:

$$P(\vec{X}|\vec{Y}) = Z_t(\vec{X}) \times \exp[G(\vec{Y})^T \times T(\vec{X})] \tag{1.7}$$

where $Z_t(\vec{X})$ is the normalization factor, $G(\vec{Y})$ is the natural parameter and $T(\vec{X})$ is the sufficient statistics of the distribution.

Another concept that we extensively refer to regarding exponential family of distributions is *conjugacy*. If $P(Y)$, the prior distribution, and $P(Y|X)$, the posterior have the same form, the prior and posterior are called conjugate distributions, and the prior is called a conjugate prior for the likelihood $P(X|Y)$. It has been shown that all exponential family distributions have conjugate priors [34].

Probabilistic models are readily convertible to learning models. The learning process is accomplished by using the training data. In choosing every probabilistic model two steps must be considered. Firstly, which model is to be chosen and secondly, how the data fit to the chosen model.

1.2.1 Model fitting and selection

In this step we assume that we have a model H which we use for fitting our data into. The model H consists of the model parameter space and the probability distribution function (PDF) that defines the distribution model. Assuming the observed data to

be $\vec{X}$ and the model parameters to be θ the Bayes theorem gives:

$$P(\vec{\theta}|\vec{X}, H) = \frac{P(\vec{X}|\vec{\theta}, H)P(\vec{\theta}|H)}{P(\vec{X}|H)} \tag{1.8}$$

In the above equation we call $P(\vec{X}|\vec{\theta}, H)$ the data likelihood and $P(\vec{X}|H)$ the Data evidence. The model fitting begins with a raw assumption of the value of the prior. After each observation our belief of the prior is updated and reflected in the posterior accordingly. In practice the process continues until a form of convergence is reached by the model.

1.3 Expectation-Maximization algorithm

The expectation-maximization (EM) algorithm is a general method for estimating the maximum-likelihood parameters of a distribution from a given data set. The algorithm works in two different instances, when the data available are incomplete or when there are missing data. The former happens when parts of the data are missing due to different reasons such as noise or occlusion. The latter case happens when optimizing the likelihood function is intractable but could further be simplified by the assumption of the existence of hidden or missing parameters. We assume that $\mathcal{X} = \left\{\vec{X_1} \dots \vec{X_N}\right\}$ is the observed data generated from an unknown distribution. We call $\mathcal{X}$ the incomplete data. Next we assume that there is a complete data set $\mathcal{Z} = (\mathcal{X}, \mathcal{Y})$ consisting of the incomplete data and hidden variables $\mathcal{Y}$. The joint distribution of the complete data is thus:

$$p(\mathcal{Z}|\Theta) = p(\mathcal{X}, \mathcal{Y}|\Theta) = p(\mathcal{Y}|\mathcal{X}, \Theta)p(\mathcal{X}|\Theta) \tag{1.9}$$

Having the new complete joint distribution function, we next proceed with defining the complete data likelihood function $\mathcal{L}(\Theta|\mathcal{Z}) = p(\mathcal{X}, \mathcal{Y}|\Theta)$.

The EM algorithm consists of two sequential steps. In the first step, the expectation, the algorithm finds the expected value of the log-likelihood of the complete

data, in respect to the unknown data space $\mathcal{Y}$:

$$Q(\Theta, \Theta^{(i-1)}) = E_y \left[\log p(\mathcal{X}, \mathcal{X}|\Theta) | \mathcal{X}, \Theta^{(i-1)} \right] \tag{1.10}$$

In above $\Theta^{(i-1)}$ is our current estimate of the model parameters. The choice of the logarithm of the likelihood instead of the likelihood offers computation convenience and it is appropriate since the logarithm function is strictly increasing.

The goal of the second step, the maximization step, is to maximize the expectation value computed in the E-step:

$$\Theta^{(i)} = \arg\max_{\Theta} Q(\Theta, \Theta^{(i-1)}) \tag{1.11}$$

The above two steps are repeated until a certain convergence criterion is met. It has been shown that the algorithm increases the log-likelihood in each step and the model converges to a local maximum [29].

1.4 Variational Inference

The EM algorithm works fine for tractable graphs. Dealing with complicated graphs with interdependencies between the nodes, it is no longer possible to use the EM algorithm for parameter estimation [88]. One of the ways for solving the intractability problem is using variational inference. Generally speaking, variational approximation proceeds with approximating the complex model with a simpler tractable model. The idea is that based on the observed data, we consider the latent variables to be independent.

The main inference problem in variational learning models is the approximation of the posterior of the hidden variables with a variational distribution such that:

$$p(\mathcal{Y} \mid \mathcal{X}) \approx \mathcal{Q}(\mathcal{Y}) \tag{1.12}$$

The key simplifying aspect of the variation method is that $\mathcal{Q}(\mathcal{Y})$ has a simpler form than the posterior $p(\mathcal{Y}|\mathcal{X})$. The goal is to ensure that the approximating function be made as similar to the true posterior as possible. The difference between $\mathcal{Q}$ and the true posterior is measured in the form of a dissimilarity function $\mathcal{D}(\mathcal{Q}, p)$. The inference algorithm hence is performed in a way so as to minimize this distance. The dissimilarity distance used in this work is the Kullback-Liebler divergence which will be explained in the next subsection.

1.4.1 Kullback-Leibler divergence

Kullback-Leibler (KL) divergence is an entropy based measure, defined as [49]:

$$KL(Q||P) = \int_y Q(y) \log \frac{Q(y)}{P(y)} dy \tag{1.13}$$

KL divergence has a srtraight forward definition in a source coding context. The divergence between Q and P is defined as the number of nats that will be wasted on average if one tries to code a distribution Q with a perfect encoder optimized for the source P [47]. Adapting the KL divergence to depict the dissimilarity between the variational approximation $Q(Y)$ and the actual posterior $P(Y|X)$ gives us the following:

$$KL(Q||P) = \int_y Q(y) \log \frac{Q(y)}{P(y|x)} dy = \int_y Q(y) \log \frac{Q(y)}{P(y,x)} dy + \log P(x) \tag{1.14}$$

In the above equation $\log p(x)$ is independent of Q and therefore minimizing the KL divergence with respect to Q is reduced to minimizing the first term of the above or equivalently maximizing the negative of it. We define $\mathcal{L}(Q)$ to be the negative of the first term:

$$\mathcal{L}(Q) = \int_Y Q(y) \log(P(y,z)dy - \int_Y Q(y) \log Q(y) dy \tag{1.15}$$

The essence of the variational models that use KL divergence for their parameter estimation is about finding proper means for maximizing the above equation so as

to minimize the divergence between the actual posterior and the variational distributions. We will use this concept later in the thesis to estimate our model parameters.

1.5 Multi-topic Models

One of the immediate applications of proper data modeling is classification. It covers a vast extend of problems such as placement of textual data into appropriate library entries or classifying objects into their relevant categories. In this context, one of the most challenging tasks is the classification of natural scenes without going deep inside their semantics. The challenge behind the former is that natural scenes are generally composed of a huge number of minute objects. The presence of these recurring objects makes it extremely complicated to develop useful classifiers based on the semantics alone. After all one would expect to see roads, trees, sun and the sky recurring in scenes both taken inside the city or in the suburb. The need to consider the presence of recurring data singletons, whether words, visual words or visual objects, led to the so-called topic based models. Latent semantic indexing (LSI) [28] is the first successful model proposed to extract recurring topics from data. It was proposed for textual documents modeling using mainly singular value decomposition (SVD). A generative successful extension of LSI called probabilistic latent semantic indexing (PLSI) was proposed in [42]. However, PLSI is only generative at the words layer and does not provide a probabilistic model at the level of documents. Therefore, two major problems arise with PLSI. Firstly, the number of parameters increases with the number of documents. Secondly, it is not clear how one can learn a document outside of the training phase. To overcome these shortcomings, the authors in [12] proposed the LDA model which has so far proven to be a reliable and versatile approach for data modeling. LDA has received a particular attention in the literature and several applications (e.g. natural scene classification [31]) and extensions have been proposed. Examples of extensions include the hierarchical version of LDA [11], used for instance in [79] for hierarchical object classification, and the online version proposed in [41]. Despite its success and elegance, the LDA model, as will be shown in the coming

chapters, has certain deficiencies. A great part of this thesis is therefore dedicated to dealing with these deficiencies by offering more realistic modeling capabilities based on the LDA model.

1.6 Hierarchical data classification

While dealing with huge amount of digital data, it is necessary to be able to efficiently classify them to relative categories. This results in ease of data retrieval and recommendation, and database browsing efficiency [74, 57]. The conventional view to object classification in machine learning is to recognize the objects inside the scene and then to categorize the scenes based on the recognized objects [82]. However, it was shown in [75] that human brain is able to categorize scenes containing objects far more quickly than the conventional object extraction model. Further, it was shown in [33] that the cerebral cortex which is responsible for the processing of visual scenes inside human brain follows a hierarchical processing model for interpreting scenes. To achieve higher speed and more accuracy, it is therefore desirable to develop models that follow the brain vision model.

In practice object classification is not an easy task due to many existing challenges like changes in illumination, scale, orientation or occlusion, each of which can have negative effect on the classification process [53]. In order to minimize the effects of these undesired factors, the natural approach is to use models which offer a level of independence from the above variations. Traditional approach to achieve this goal is to use global features of the image data such as color and texture as the basis for classification. However, another approach which has received more popularity recently is to find low level features inside different image classes and to use them as class interpreters. Low level local features properly applied can offer an acceptable level of robustness towards traditional vision challenges such as scaling, changes of illumination and occlusion [53]. In this thesis we focus on low level local features of image data.

After the features are extracted from the image, they should be interpreted to be able

to discriminate different classes and categories. To this aim Csurka et al. proposed the bag of visual words model [27]. The goal of the bag of visual words model is to describe an image as a set of predefined visual words. In this perspective bag of visual words model gives a model conceptually similar to lingual vocabulary. Whilst in language a predefined set of words exists for describing the language, in bag of visual words model, a set of basic visual words is defined to describe the image data space. As shall be shown an optimal choice of the visual words and their number can greatly improve the model accuracy. Following the introduction of the bag of visual words model several authors have proposed their improvements to the original model e.g. [89] [30] [62]. One common point among most of the new models is that, nearly all of them make use of the existing text classification models and adopt them to the image data.

In many applications, it is convenient to classify the data in a hierarchical form [1]. This way the more general classes merge together to form the parents to the more specific ones. One can look at this approach like the book placement strategy inside a library. Whilst in a flat classification one has to look through every different class to find what one is looking for, in a correctly arranged library one knows that for instance he can find a circuit analysis book, in the electrical engineering section, which is in return in the engineering section and so on. One obvious point in developing hierarchical based models is that the model itself must be hierarchy adaptable. One of the most referred of these family of models is the latent Dirichlet allocation (LDA) model [12] and its hierarchical adaptation [10]. Based on the hierarchical LDA model Fei-Fei et al. proposed their supervised hierarchical model for scene classification [32]. Sivic et al. developed the hierarchical LDA model further to propose an unsupervised model for finding class hierarchies [79].

Another model that can be used for hierarchical classification is the hierarchical Dirichlet model recently developed in [86] for document classification. In this thesis we propose adopting this model, for hierarchical object classification purposes. As it will be shown, as expected, the hierarchical nature of the model results in considerable improvement in image database classification accuracy. The other advantages of

the method are its ease of implementation and low computation load. As the name of the above model implies, it considers the Dirichlet distribution, as an integrated basis. However, as it has been mentioned in previous works like [58], Dirichlet distribution may not be the best choice as a prior in statistical models. Subsequently we propose a new hierarchical model as a generalization to the model proposed in [86] and adopted in [2]. In this model we extend the hierarchical Dirichlet model of [86] by considering the more flexible generalized Dirichlet distribution. It is our assumption that the more generalized covariance matrix of the generalized Dirichlet distribution in comparison with Dirichlet distribution, which has a very restrictive negative covariance, results in more efficient and realistic data modeling. As an elaboration on this idea one may think of the text modeling application and the lingual vocabulary. It is expectable to assume that in the class of philosophy there is a positive correlation between the occurrences of the words Greece and philosophy, however if we take the Dirichlet distribution as the basis for our word generation model the model always unconditionally assumes a negative correlation between the two words which is not what is expected.

One prior distribution that has recently gained notice in count data modeling is the Beta-Liouville (BL) distribution [38]. The BL distribution offers comparably better modeling characteristics without adding much complexity to the models, while avoiding overfitting and high sensitivity observed in previous models [3]. Thus, we also proceed with the idea of replacing the Dirichlet distribution with BL distribution as proposed in [16] with the idea of using the hierarchical structure for improving the classification accuracy in [86] to derive a new hierarchical model which would enhance the results of [86]. The BL distribution offers a more versatile covariance matrix than the Dirichlet distribution, as will be shown later, while requiring only slight increase in computational requirements in comparison to the Dirichlet distribution. This allows the hierarchical model to act as a trade off between the Dirichlet simplicity and and generalized Dirichlet strong modeling efficiency.

To better understand the concept of hierarchical classification it is helpful to consider the visual hierarchy in figure 1-2. Notice how by moving down towards the

lower branches of the database the hierarchy becomes more object oriented.

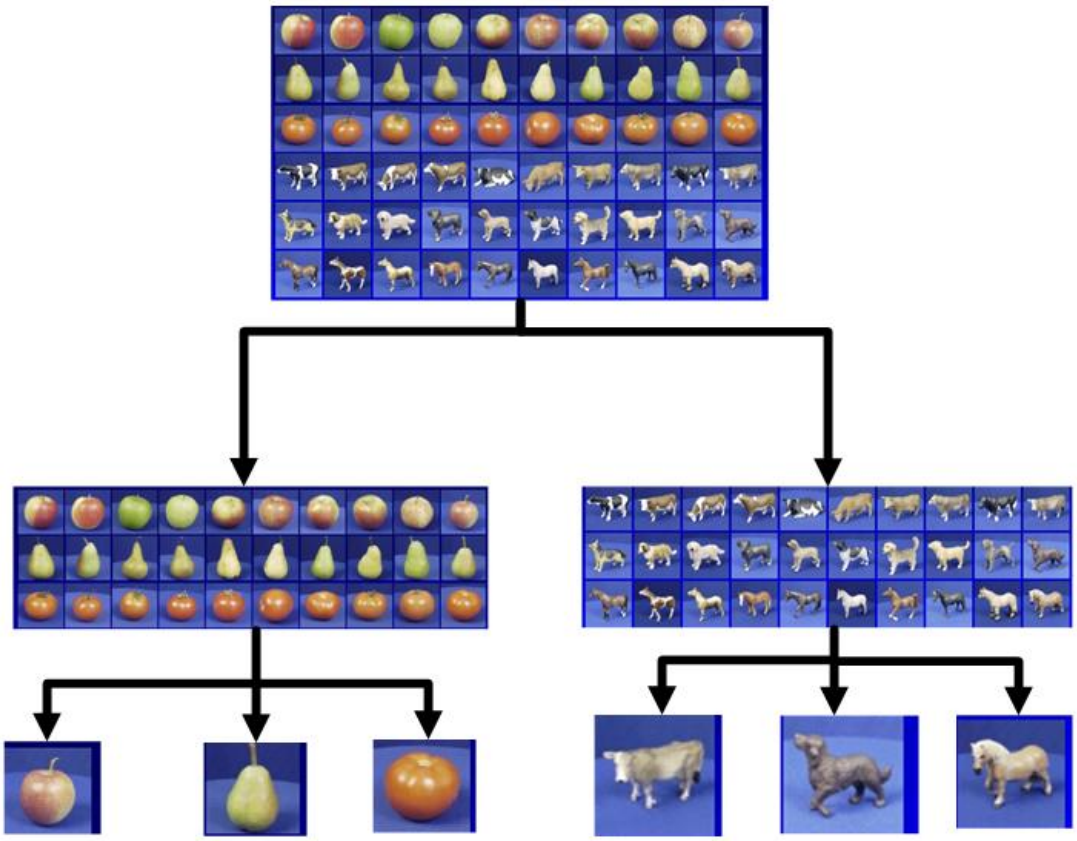


Figure 1-2: An example of hierarchical object classification. Note how by moving down the hierarchy each of the nodes become more categorized.

1.7 List of contributions

The several contributions of this thesis were either published or being reviewed in different high reviewed scientific Journals and conferences as of the time of the preparation of the thesis. The list of the contributions are as follows:

1. Designing a hierarchical statistical model for object classification using Dirichlet distribution and showing the relative strength of the model in comparison to naive Bayes model [2]. Designing a hierarchical statistical framework for count data modeling using generalized Dirichlet distribution [3]. Designing

a hierarchical statistical model for count data modeling using Beta-Liouville distribution[4].

2. Designing a semisupervised online learning hierarchical structures for object classification and developing three distinct online learning models using Dirichlet, generalized Dirichlet and Beta-Liouville distributions[5].

3. Designing a variational Bayes model for count data learning and classification using generalized Dirichlet distribution as an improvement to the LDA model [8].

4. Designing a latent topic model based on the Beta-Liouville distribution as an improvement to the LDA model with more computational efficiency in comparison to the previous contribution[6].

5. Designing online learning for the last two latent topic models [7].

1.8 Thesis Structure

The organization of the thesis is as follows. In chapter 2 we introduce the static hierarchical models and their application in object classification. In chapter 3 we extend our hierarchical models to include auto learning hierarchal structures. In chapter 4 we introduce our approach to multi-topic models and their applications in text, object and scene classification as well as action recognition and we shall describe our proposed online learning scheme. Finally, in chapter 5 we shall conclude the thesis.

Chapter 2

A Hierarchical Statistical Model For Object Classification

The problem that we address in this chapter is that of learning hierarchical object categories. Object classification in computer vision can be looked upon from several different perspectives. From the structural perspective object classification models can be divided into flat and hierarchical models. Many of the well-known hierarchical structures proposed so far are based on the Dirichlet distribution. In this chapter, we present three distinct works. The first one is an adaptation of the hierarchical Dirichlet model proposed in [86] to work for the object classification and categorization application. The results of the adaptation were published in [2]. The second work presented in this chapter considers the structural model of [86] and adapts the generalized Dirichlet as the prior assumption. The results of this work were published in [3]. The third work in this chapter proceeds with finding a means for combining the simplicity of the first model and the efficiency of the second one through the introduction of the Beta-Liouville distribution and its adaptation to the hierarchical model. The results of this adaptation were published in [4].

2.1 The Hierarchical Dirichlet Model

Let $Im = \{Im_1, ..., Im_N\}$ be the set of the images to be categorized. Following the guidelines of [27], the first step for categorization is to detect low level features from image data. In this step a proper feature detector is applied on each image Im_n and the desirable features are extracted. In the next step, each of the extracted descriptors from the last step is assigned to the nearest predetermined visual word. In this chapter the Euclidean distance is considered as the measure of distance between vectors. The next step is the bag of visual words construction. In this step a frequency vector of the assigned descriptors from the last step is constructed. The dimension of the vector is the same as the number of the visual words in the vocabulary V and the value of each of the vector elements is equal to the number of times the according descriptor is observed inside the image. We denote the set of the frequency vectors constructed as above as $I = \{I_1, ..., I_N\}$, where each I_i is the constructed frequency vector for the image Im_i.

The approach proposed in [27] is based on the naive Bayes model. It considers the class conditional probabilities of the occurrence of each of the words in different classes and then uses the multinomial distribution to find the class which gives the highest posterior probability to the present frequency vector. One serious drawback with the above model is that it refuses to see any interdependency between the occurrences of the visual words. The only condition which holds on the probability distribution of the visual words is

$$\sum_{k=1}^{K} P(w_i) = 1 \tag{2.1}$$

where $w = (w_1, ..., w_K)$ is the vector of the visual words. Yet, one can rightfully assume that the occurrences of visual words are not totally independent of each other. However, the above model fails to realize the concept of hierarchical vision, since in the naive Bayes model the classes are defined separately from each other. In order to solve the above problems, we propose using the hierarchical Dirichlet model, instead of the Naive Bayes model, which we describe in the following.The model proposed in [27] assumes a flat multinomial generative model which works under the

assumption that the data are generated from a generative model with class parameter $\theta_i = (\theta_{i1}, ..., \theta_{iK})$.

$$p(I_n|\theta_i) \propto \frac{(\sum_{k=1}^{K} I_{n_k})!}{\prod_{k=1}^{K} I_{n_k}!} \prod_{k=1}^{K} (\theta_{ik})^{I_{n_k}} \tag{2.2}$$

where I_{nk} is the number of occurrences of the *k-th* visual word inside image Im_n. The unknown here is the parameter vector θ_i, which should be estimated from the training data. Using the occurrences alone gives generally poor estimates [19]. An appropriate solution to address this issue is the introduction of a prior information into the construction of the statistical model. The prior information, for the multinomial assumption, is chosen in general to be given by the Dirichlet distribution [19, 61]. A better approach has been recently proposed in [86] for the hierarchical classification of text documents. This approach which, unlike the previous flat model (equation 2.2), takes into account the notion of hierarchy by selecting a hierarchical Dirichlet distribution, inside the hierarchy, for the parameter vector θ_i, meaning:

$$\theta_i \sim \begin{cases} \mathcal{D}(\eta, ..., \eta) & \text{if } i \text{ is the first node in the hierarchy} \\ \mathcal{D}(\sigma \times \theta_{pa(i)}) & \text{for the other nodes in the hierarchy} \end{cases} \tag{2.3}$$

where $\theta_{pa(i)}$ is the parent node of the *ith* node and $\mathcal{D}$ represents a Dirichlet distribution. Note that a Dirichlet distribution with parameter vector $(\alpha_{i1}, ..., \alpha_{iK})$ is defined over the hyper plane $\sum_{k=1}^{K} \theta_{ik} = 1$, as [67]:

$$p(\theta_{i1}, ...\theta_{iK}) = \frac{\prod_{k=1}^{K} \Gamma(\alpha_{ik})}{\Gamma(\sum_{k=1}^{K} \alpha_{ik})} \prod_{k=1}^{K} \theta_{ik}^{(\alpha_{ik}-1)} \tag{2.4}$$

where Γ is the Gamma function. The hierarchical nature of the model gives the following useful relationship [86]

$$E[\theta_i|\theta_{pa(i)}] = (\sigma \times \theta_{pa(i)})/(\sum_{k=1}^{K} \sigma\theta_{pa(i)}(k)) = \theta_{pa(i)} \tag{2.5}$$

It is noteworthy that generally the hyper parameter η is set to 1. In the above σ is the hierarchy inheritance parameter. The higher values of σ leads to tighter bound

between the parent and children nodes. As it will be shown in later sections the correct choice of this hierarchical bound is an important factor in model efficiency. In this chapter, we choose the optimum value of σ experimentally, as the value that maximizes the model classification success rate. A rather more analytical method for calculating σ is described in [86].

Parameters Estimation

In order to estimate the parameters of the model, we consider that a class based training data is available. Assuming $n_i = (n_{i1}, ...n_{iK})$ be the vector obtained by adding up all the frequency vectors in class i. Therefore, the value $N_i = \sum_j n_{ij}$ is actually the total number of visual word occurrences inside the training set of class i. Since n_i itself can be considered as a meta image defined by the θ_i parameters, n_i follows a multinomial distribution defined by θ_i. An interesting characteristic of the Dirichlet distribution is its conjugacy to the multinomial distribution. This implies that:

$$p(\theta_i|n_i) \propto \mathcal{D}(\alpha_{i1} + n_{i1}, ..., \alpha_{iK} + n_{iK}) \tag{2.6}$$

Considering the conjugacy between the Dirichlet and multinomial distribution it is easy to show that [19]

$$\hat{\theta}_i = \frac{n_i + \sigma\hat{\theta}_{pa(i)}}{N_i + \sigma} \tag{2.7}$$

A possible parameters estimation approach is the one proposed in [86] based on linear minimum mean squared error estimate (LMMSE). LMMSE estimator for the node θ_i with m child node is computed as

$$\theta_i = E[\theta_i] + M^{-1} \times \begin{pmatrix} \hat{\theta}_{ch(i1)} - E[\theta_i] \\ \hat{\theta}_{ch(i2)} - E[\theta_i] \\ . \\ \hat{\theta}_{ch(im)} - E[\theta_i] \end{pmatrix} \tag{2.8}$$

where

$$M = \begin{pmatrix} \Sigma(\theta_{ch(i1)}) & \Sigma(\theta_{ch(i1)}\theta_{ch(i2)}) & \dots & \Sigma(\theta_{ch(i1)}, \theta_{ch(im)}) \\ \Sigma(\theta_{ch(i2)}, \theta_{ch(i1)}) & \Sigma(\theta_{ch(i2)}) & \dots & \Sigma(\theta_{ch(i2)}, \theta_{ch(im)}) \\ \vdots & \dots & \dots & \vdots \\ \Sigma(\theta_{ch(im)}\theta_{ch(i1)}) & \dots & \dots & \Sigma(\theta_{ch(im)}) \end{pmatrix}$$

and $\Sigma\theta_i$ is the variance of θ_i. As was shown in [86] the non diagonal elements of the estimation matrix are all equal to $\Sigma\theta_i$. This exchangeability of the parameters allows the above equation to be expressed in a much simpler form:

$$\hat{\theta}_i = \frac{\sigma\hat{\theta}_{pa(i)} + m(\sigma+1)\bar{\hat{\theta}}_{ch(i)} + n_i}{\sigma + m(\sigma+1) + N_i} \tag{2.9}$$

In the above equation, $\bar{\hat{\theta}}_{ch(i)} = \frac{1}{m}\sum_{j \in ch(i)} \hat{\theta}_j$, is the average of the node children. It's interesting to note also that for the leaf nodes the above equation reduces to:

$$\hat{\theta}_i = \frac{\sigma\hat{\theta}_{pa(i)} + n_i}{\sigma + N_i} \tag{2.10}$$

which is actually a simple update equation. In order to exploit the update equation, the θ parameters are initialized by hierarchical Dirichlet samples, and the above iteration equation is calculated until a convergence criterion is met. One problem that remains, however, is the choice of the value of σ. In our work we use the training data to optimize the choice of σ. The effect of choosing different values for σ is brought in the following section.

2.1.1 Experimental Results

In this section we illustrate and discuss the proposed statistical model for an image database hierarchical classification task.

Image Dataset

We chose the ETH-80 dataset [51] as the basis for our hierarchical classification. The main reason for this choice is that the dataset is optimized for object classification

purposes. The dataset contains views and segmentation masks of 80 objects, each one photographed in more than 40 different poses. In total the database contains more than 3000 images. The objects are classified in 8 categories cows, dogs, horses, apples, pears, tomatoes, cups, and cars, from which we choose the 7 first categories for our hierarchical classification. The reason for choosing the first seven categories

Figure 2-1: Samples of the ETH-80 dataset [51].

is that 6 of them can be classified in 2 unique categories, fruits and animals. As it will be shown in the results section the visual similarities between the chosen classes contribute much to the efficiency of the hierarchical classification, since one may expect quite similar visual words to be extracted from visually similar classes. Approximately 20 percent of the image database is randomly chosen as the training database, whilst the remaining images form our test dataset.

Feature Extraction and Visual Words Generation

We use the Scale invariant feature transform (SIFT) descriptors [53] for feature extraction from our images. The choice of SIFT descriptors over the other available descriptors is due to several factors. The high dimensionality of the SIFT descriptors and its comparably robustness towards changes in scaling, illumination, occlusion, etc, compared with other feature descriptors, results in better discrimination between the extracted descriptors [55] . The next step is the generation of the visual vocabulary (the set of visual words) from the entire dataset. Assuming for the moment that we have access to the entire unclassified dataset, we generate our visual vocabulary as

follows. We derive the entire set of descriptors by running the SIFT algorithm over the entire dataset. This operation leads to a set of descriptors. In the next step we apply a clustering algorithm, in this work the K-means algorithm, over the obtained set to find the descriptors centroides. We assign each derived centroides to a visual words and call the collection of the visual words the visual vocabulary. One interesting point here is that the number of centroides is actually arbitrary, and as we show later an optimal choice of K can lead to optimum results. Since the proposed hierarchical model is in fact the hierarchical generalization of the naive Bayes model, in the experiments we compare the efficiency of our model against the naive Bayes model.

Assumed Hierarchical Structure

In order to show the effect of the hierarchical model we propose the hierarchical structure shown in figure 2-2 for our classification. The choice of the hierarchy structure is

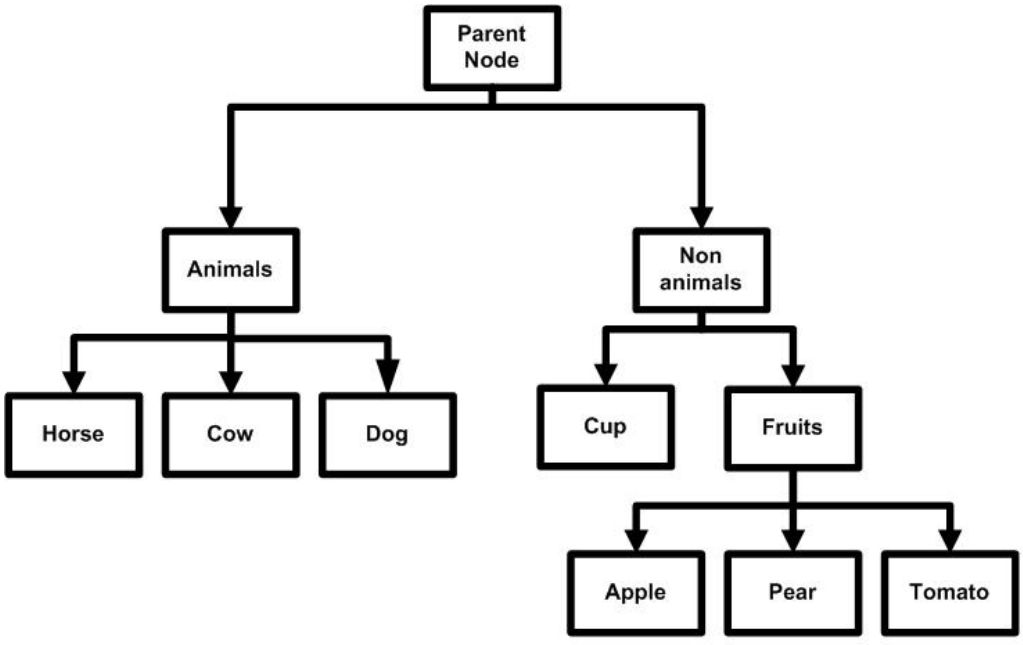


Figure 2-2: The Hierarchical structure chosen for the image database classes. The choice of the hierarchy elements is based both on visual and conceptual similarities between the classes.

based on the visual similarities between the classes. However, in order to emphasize

the importance of visual similarity in our hierarchical classification we also use a visually irrelevant class, the cups, in our model and we will show that a poor selection of hierarchies can contribute little if any to the model quality.

Analysis of the Recognition Capability of the Model

In order to analyze the recognition capability of the model, a series of experiments are performed. For the object recognition purposes the lowest leaves of the tree, which contain individual objects, are chosen for analysis. As was shown in the previous section, two parameters directly contribute to the efficiency of the model i.e. the number of visual words and the value of σ. We define the model success rate as the ratio between the total number of the correctly classified images in all classes against the total number of images. Figure 2-3 shows the recognition success rate of the model as a function of the number of visual words. As it can be seen from this figure, the model recognition success rate reaches its maximum around 300 visual words. It must be noted that in figure 2-3 the outcome for number of visual words more than 500, is subject to over training and the number of multiple recognitions for a simple object becomes unacceptably high as the number of chosen words increases. Moreover, the effect of increasing the hierarchy strength can be seen, by increasing the value of σ, in sub figures. In the top left figure, with a small value of σ, the model behaves only slightly differently from the naive Bayes model, but as the value of σ increases the model becomes more hierarchical based and drifts away from the naive Bayes model. It can also be seen from the figure that compared with the naive Bayes model the model in general reaches higher success rates.

Figure 2-4 shows the effect of choosing different values of sigma on the model recognition success rate. It also shows that recognition success rate increases slightly by the increase of the sigma value from 1 to 100, in the acceptable visual word range of 200 to 500 words, letting us conclude that for our database the recognition is optimized for $\sigma = 100$ and $K = 300$. One may notice, however, that the model in average has a 50% success rate in object recognition. This low success rate is, to much extend, because the hierarchy works in the way that assigns quite similar θ parameters to

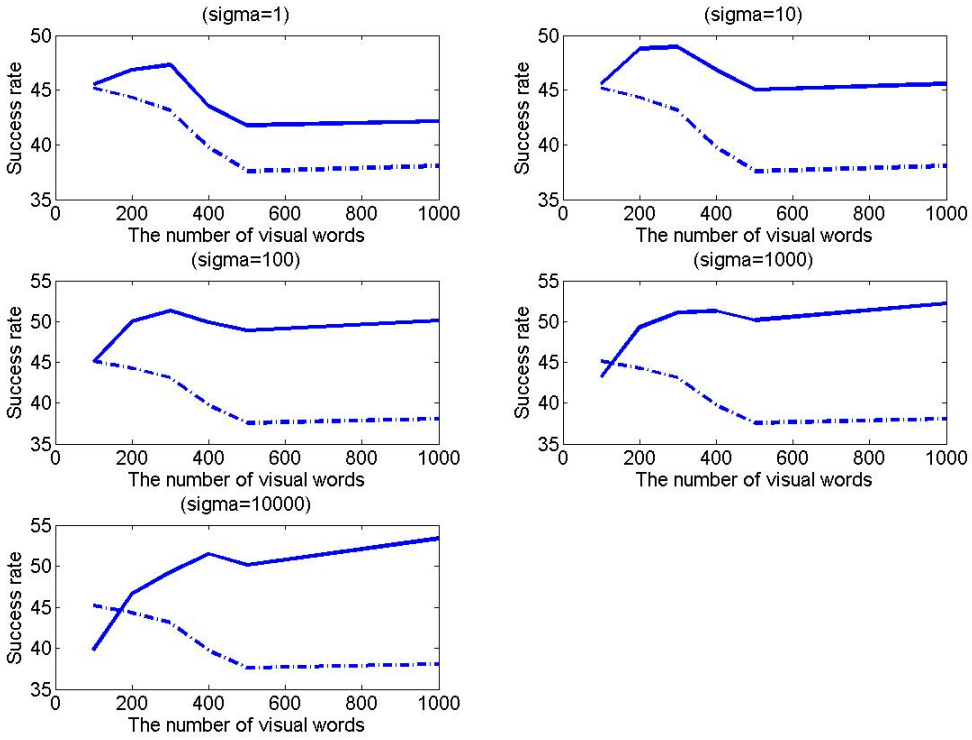


Figure 2-3: The solid line shows the model recognition success rate versus the number of visual words for different values of sigma. The dashed line shows the success rate for the naive Bayes model under the same condition.

the lowest nodes which define the individual objects. This can be mathematically seen from equation 2.10. Thus, the system is unable to discriminate visually similar objects efficiently. However, as we move up inside the hierarchy and the node parameters become not only dependant on the parent nodes but also their children nodes, as we can see from equation 2.9, the success rate of the model improves. This, in general, is in accordance with the assumption that the model classification improves throughout the hierarchy. It will be shown in the next section that the model behaves as expected as we move upper in the hierarchy.

Analysis of the Categorization Capability of the Model

As it was mentioned in the previous sections the main idea of the model is to act as an efficient classifier. Should the model behave as we expect, going further up in the

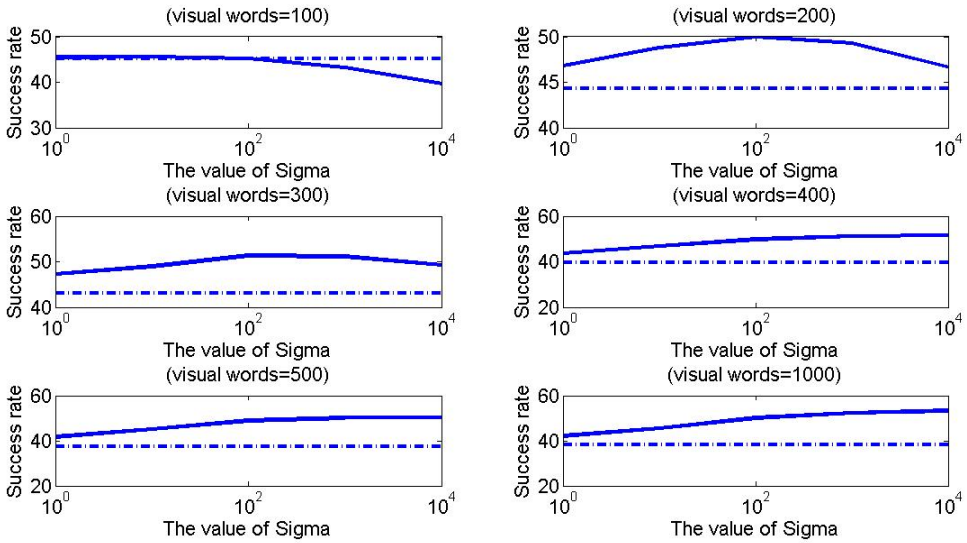


Figure 2-4: The solid line shows the model recognition success rate versus sigma for different number of visual words. The dashed line shows the success rate for the naive Bayes model under the same condition.

hierarchy shall result in better recognition success rate. Figure 2-5 shows the average success rate for the second tier of the hierarchy versus the number of the visual words for different values of sigma. The data is generated by averaging the success rate of the second tier categories. As expected the model shows a considerable jump in classification accuracy by taking one step up in the hierarchy. Again it can be seen from the figure that, similar to the case of object recognition, the system reaches its maximum accuracy at around 300 visual words with sigma value equals to 100.

Figure 2-6 shows the effect of choosing different values of sigma for the second tier categorization success rate. Again it can be seen from figure 2-6 that the model success rate improves slightly by the optimum choice of sigma. The obtained improvement in the second tier is due to the right choice of the hierarchical structure based on visual word similarities of the objects. One point that needs to emphasized here, is

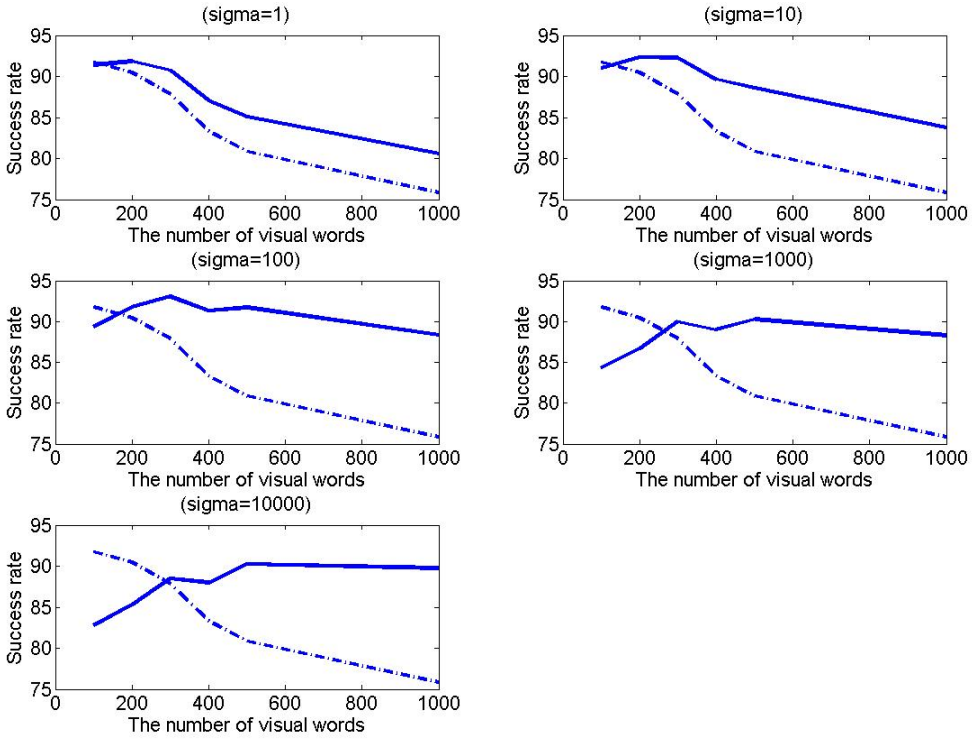


Figure 2-5: The solid line shows the model second tier categorization success rate versus the number of visual words for different values of sigma. The dashed line shows the success rate for the naive Bayes model under the same condition.

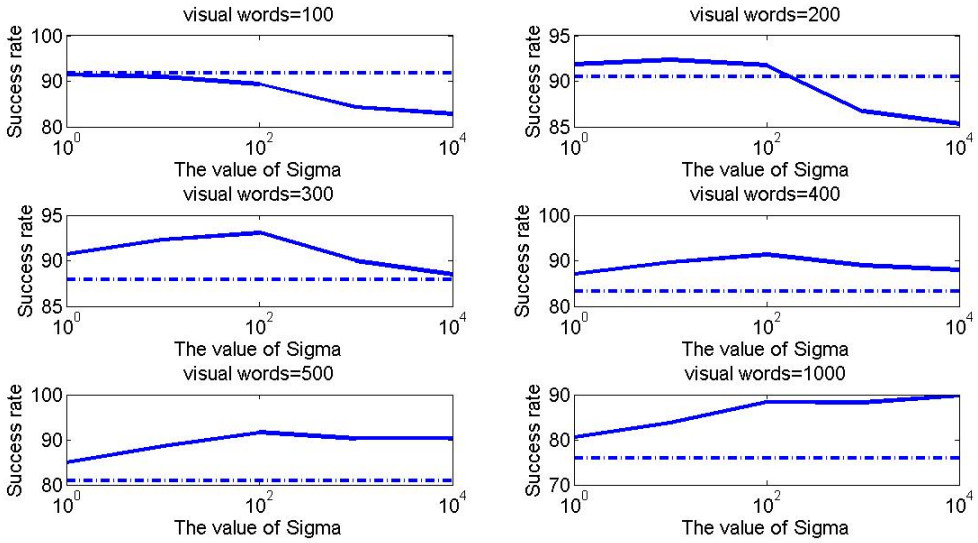


Figure 2-6: The solid line shows the model second tier categorization success rate versus sigma for different number of visual words. The dashed line shows the success rate for the naive Bayes model under the same condition.

that increasing the number of visual words does not necessarily lead to improved success rate. The reason behind this observation is that once one begins to increase the number of visual words beyond an optimal number, the visual words tend to capture minute characters of the objects as well as possible clutter. This in return leads to a total drop in the model accuracy as can be seen from figure 2-6 In order to show that the wrong choice of elements inside the hierarchy contributes little if any to the system accuracy, we show the effect of misplaced class of objects, cups, on the system accuracy.

Table 2.1 shows the confusion matrix generated by optimal choice of system parameters. As it can be seem from table 2.1, the correct choice of the hierarchy elements

Table 2.1: Confusion matrix of the model for $\sigma = 100$ and 300 visual words.

Class	*apple*	*cow*	*cup*	*dog*	*horse*	*pear*	*tomato*
apple	57.6	0	16.4	1.4	1.4	7.5	19.9
cow	0	21	8.3	6.3	15.6	0	3.1
cup	5.2	0	48.2	1.4	0	1.1	0
dog	0	26.3	5.7	38.1	20.8	0.8	5.7
horse	0.5	51.4	12.1	50.5	60.4	0.8	13.8
pear	27.1	0	6.9	1.4	1.7	88.7	11.2
tomato	9.3	0	2	0.5	0	0.8	45

leads to possible errors that are compensated in the upper tiers, whilst the bad choice of the elements leads to the errors which propagate into the upper levels of hierarchy. To better understand this point, one should note the confusion matrix for the class 'cup'. Even though the class is classified as non-animals, it can be readily seen that the classification errors inside the cup class are not confined to neighboring nodes, therefore putting 'cup' under non-animal hierarchy, though conceptually correct, is visually not a correct choice.

Having shown the merit of using the hierarchical Dirichlet model for object classification, we proceed with proposing our first improved model using the generalized Dirichlet distribution. In the next section to follow we thus introduce the hierarchical generalized Dirichlet model and we make a thorough comparison between the hierarchical Dirichlet and the hierarchical generalized Dirichlet model.

2.2 Hierarchical generalized Dirichlet Model

In the last section we showed the feasibility of adapting hierarchical learning models for object classification. In this chapter we proceed with offering a new hierarchical scheme based on the generalized Dirichlet distribution as a replacement for the previous Dirichlet assumptions. We make a thorough comparison between the two models and analyze the merits and the drawbacks of the new model.

further to propose an unsupervised model for finding

Generalized Dirichlet as a Prior

If a random vector $\vec{\theta}_I = (\theta_{I1}, \ldots, \theta_{IK})$ follows a generalized Dirichlet distribution with parameters $\xi = (\alpha_{I1}, \beta_{I1}, \ldots, \alpha_{IK}, \beta_{IK})$, the joint probability density function (pdf) is given by:

$$p(\vec{\theta}_I|\xi) = \prod_{i=1}^{K} \frac{\Gamma(\alpha_{Ii} + \beta_{Ii})}{\Gamma(\alpha_{Ii})\Gamma(\beta_{Ii})} \theta_{Ii}^{\alpha_{Ii}-1} (1 - \sum_{j=1}^{i} \theta_{Ij})^{\gamma_{Ii}} \tag{2.11}$$

where $0 < \theta_{Ii}$ and $\sum_{i=1}^{K} \theta_{Ii} < 1$ for $i = 1, \ldots, K$, and $\gamma_{Ii} = \beta_{Ii} - \alpha_{I(i+1)} - \beta_{I(i+1)}$. It must be noted that when $\beta_{Ii} = \alpha_{I(i+1)} + \beta_{I(i+1)}$ the generalized Dirichlet distribution is reduced to Dirichlet distribution. The mean and the variance of the generalized Dirichlet distribution with the above parameters are as follows:

$$E(\theta_{Ii}) = \frac{\alpha_{Ii}}{\alpha_{Ii} + \beta_{Ii}} \prod_{k=1}^{i-1} \frac{\beta_{Ik}}{\alpha_{Ik} + \beta_{Ik}} \tag{2.12}$$

$$Var(\theta_{Ii}) = E(\theta_{Ii}) \left(\frac{\alpha_{Ii} + 1}{\alpha_{Ii} + \beta_{Ii} + 1} \prod_{k=1}^{i-1} \frac{\beta_{Ik} + 1}{\alpha_{Ik} + \beta_{Ik}} + 1 - E(\theta_{Ii}) \right) \tag{2.13}$$

and the covariance between θ_{Ii} and θ_{Ij} is given by:

$$Cov(\theta_{Ii}, \theta_{Ij}) = E(\theta_{Ij}) \times \left(\frac{\alpha_{Ii}}{\alpha_{Ii} + \beta_{Ii} + 1} \prod_{k=1}^{i-1} \frac{\beta_{Ik} + 1}{\alpha_{Ik} + \beta_{Ik}} + 1 - E(\theta_{Ii}) \right) \tag{2.14}$$

Other interesting properties and applications of the distribution can be found in [58, 60, 59, 68]. As it can be seen from equation 2.14, the covariance of the generalized Dirichlet distribution has a more general form than the Dirichlet distribution and it is therefore possible for two random variables inside the random vector to be positively correlated [64, 65]. An interesting character of the generalized Dirichlet distribution is that, like the Dirichlet distribution, it is a conjugate prior to the multinomial distribution [66, 63]. The conjugacy between the two distributions implies that if $\vec{\theta}_I$ follows a generalized Dirichlet distribution with parameters $(\alpha_{I1}, \ldots, \alpha_{IK}, \beta_{I1}, \ldots, \beta_{IK})$ and $\vec{n}$, as defined in the last subsection, follows a multinomial with parameter $\vec{\theta}_I$, then it can be shown that the posterior distribution also has a generalized Dirichlet distribution $\vec{\theta}|\vec{N} \propto GD(\alpha_{I1}^{'}, \ldots, \alpha_{IK}^{'}, \beta_{I1}^{'}, \ldots, \beta_{IK}^{'})$ with the following parameters [90, 58]:

$$\alpha_{Ii}^{'} = \alpha_{Ii} + n_i \tag{2.15}$$

$$\beta_{Ii}^{'} = \beta_{Ii} + \sum_{l=i+1}^{K+1} n_{Il} \tag{2.16}$$

We shall use this property of the generalized Dirichlet distribution, for deriving the parameters estimation, based on the observed training data in the following subsections.

Hierarchical generalized Dirichlet Model

In our model we follow the same count data generation approach as the hierarchical Dirichlet model with the exception that the parameter vectors $\vec{\theta}_I$ have generalized Dirichlet distributions with the following special hierarchical parameters definition:

$$\vec{\theta}_I \sim \begin{cases} \mathcal{GD}(\eta, \ldots, \eta, \zeta, \ldots, \zeta) & \text{if } I \text{ is the first node} \\ \mathcal{GD}((f(\vec{\theta}_{pa(I)}), g(\vec{\theta}_{pa(I)}))) & \text{otherwise} \end{cases} \tag{2.17}$$

where $\vec{\theta}_{pa(I)}$ indicates the parent of the $I-th$ node. The functions $f(\vec{\theta}_{pa(I)})$ and $g(\vec{\theta}_{pa(I)})$ depend on the parent node and they must be determined in the way that

holds the following relationship:

$$E[\vec{\theta_I}|\vec{\theta}_{pa(I)}] = \vec{\theta}_{pa(I)} \tag{2.18}$$

By defining $\vec{f_I}$ and $\vec{g}_I$ functions as $\vec{f_I} = \{f_{I1}, \ldots, f_{IK}\}$ and $\vec{g}_I = \{g_{I1}, \ldots, g_{IK}\}$ and using equation 2.18, it can be shown by induction that (See Appendix a)

$$g_I(k) = \frac{(1 - \sum_{l=1}^{k} \theta_{pa(I)}(l))}{\theta_{pa}(k)} f_I(k) \tag{2.19}$$

One interesting fact that can be derived from equation 2.19 is that, by choosing linear dependency between $\vec{f_I}$ and $\vec{\theta}_{pa(I)}$, that is $\vec{f_I} = \sigma\vec{\theta}_{pa(I)}$ and by proper replacement in equation 2.19 one has (see Appendix b):

$$g_{I(i)} = f_{I(i+1)} + g_{I(i+1)} \tag{2.20}$$

and therefore, the hierarchical generalized Dirichlet distribution is reduced to hierarchical Dirichlet distribution. Thus, our model generalizes the hierarchical model in [86]. Like the case of the hierarchical Dirichlet model we assume the values of η and ζ to equal unity.

Following the assumptions of [86] we propose to use a LMMSE estimator for estimating the model parameters. The LMMSE estimation for the parameter $\vec{\theta_I}$ is given as:

$$\theta_i = E[\theta_i] + M^{-1} \times \begin{pmatrix} \hat{\theta}_{ch(i1)} - E[\theta_i] \\ \hat{\theta}_{ch(i2)} - E[\theta_i] \\ \cdot \\ \hat{\theta}_{ch(im)} - E[\theta_i] \end{pmatrix} \tag{2.21}$$

where

$$M = \begin{pmatrix} \Sigma(\theta_{ch(i1)}) & \Sigma(\theta_{ch(i1)}\theta_{ch(i2)}) & \ldots & \Sigma(\theta_{ch(i1)}, \theta_{ch(im)}) \\ \Sigma(\theta_{ch(i2)}, \theta_{ch(i1)}) & \Sigma(\theta_{ch(i2)}) & \ldots & \Sigma(\theta_{ch(i2)}, \theta_{ch(im)}) \\ \vdots & \ldots & \ldots & \vdots \\ \Sigma(\theta_{ch(im)}\theta_{ch(i1)}) & \ldots & \ldots & \Sigma(\theta_{ch(im)}) \end{pmatrix}$$

where $\Sigma(\vec{\theta}_{ch(Ik)}, \vec{\theta}_{ch(Ij)})$ is the correlation matrix between the parameter vectors of the $k-th$ and $j-th$ children of the $I-th$ node.

As it was shown in [86], assuming equation 2.18 to hold, one can derive the following relationships between parental and children nodes:

$$E[\vec{\theta}_{ch(I)}] = E[\vec{\theta}_I] \tag{2.22}$$

$$\Sigma(\vec{\theta}_I, \vec{\theta}_{ch(Ij)}) = \Sigma(\vec{\theta}_I) \tag{2.23}$$

$$\Sigma(\vec{\theta}_{ch(Ij)}, \vec{\theta}_{ch(Ik)}) = \Sigma(\vec{\theta}_I) \tag{2.24}$$

$$\Sigma(\vec{\theta}_{ch(Ik)}) = \Sigma\vec{\theta}_I + E_{\vec{\theta}_I}[\Sigma(\vec{\theta}_{Ik}|\vec{\theta}_I)] \tag{2.25}$$

According to Bayes theory and based on the existing conjugacy between the multinomial and generalized Dirichlet distribution, it can easily be shown that the variance of each of the elements of the θ_i can be derived as:

$$\begin{aligned} var(\theta_{Ij}|\vec{n}_I) = \quad & E[\theta_{Ij}]\Bigg(\frac{f_{Ij}+n_{Ij}+1}{f_{Ij}+n_{Ij}+g_{Ij}+\sum_{l=j+1}^{K+1} n_{Il}} \prod_{w=1}^{j-1} \frac{g_{Iw}+\sum_{y=w+1}^{j-1} n_{Iy}+1}{f_{Iw}+n_{Iw}+g_{Iw}+\sum_{y=w+1}^{k+1} n_{Iy}} + \\ & 1 - E[\theta_{Ij}]\Bigg) \end{aligned} \tag{2.26}$$

and the covariance between $k-th$ and $j-th$ element of θ_I is derived as:

$$\begin{aligned} cov(\theta_{Ik}, \theta_{Ij}|\vec{n}_I) = E[\theta_{Ij}]\Bigg(& \frac{f_{Ik}+n_{Ik}+1}{f_{Ik}+n_{Ik}+g_{Ik}+\sum_{l=k+1}^{K+1} n_{Il}} \\ & \prod_{w=1}^{k-1} \frac{g_{Iw}+\sum_{y=w+1}^{k-1} n_{Iy}+1}{f_{Iw}+n_{Iw}+g_{Iw}+\sum_{y=w+1}^{k+1} n_{Iy}} + 1 - E[\theta_{Ik}]\Bigg) \end{aligned} \tag{2.27}$$

Having the above equations we derive the covariance matrix of θ_I as:

$$\sum \vec{\theta}_I|\vec{n}_I = diag\left[var(\vec{\theta}_{Ij}|\vec{n}_I)\right] + cov\left[(\vec{\theta}_I|\vec{n}_I)\right] \tag{2.28}$$

where $diag(var(\vec{\theta}_{Ij}|\vec{n}_I))$ is the diagonal matrix of the variances of the elements of $\vec{\theta}_I$ and $cov(\vec{\theta}_I|\vec{n}_I)$ is the covariance matrix of $\vec{\theta}_I$. The last required equation to be derived

analytically is the parent conditional expectation of the covariance matrix of the parameter $\vec{\theta}_I$. In order to find an analytic equation for this parameter we must make some simplifying assumptions which we will describe in the following relationships. Analytically for the parent conditional expectation of variance we have:

$$E_{\vec{\theta}_{pa(i)}}[var(\theta_{Ij})|\vec{\theta}_{pa(I)}] = E_{\vec{\theta}_{pa(I)}}\Bigg(\theta_{pa(Ij)} \times [\frac{f_{Ij}+1}{f_{Ij}+g_{Ij}} \prod_{w=1}^{j-1} \frac{g_{Iw}+1}{f_{Iw}+g_{Iw}} + 1 - \theta_{pa(Ij)}]\Bigg) \tag{2.29}$$

and the parental conditional expectation of the covariance between $k-th$ and $j-th$ element of $\vec{\theta}_i$ is derived as:

$$E_{\vec{\theta}_{pa(I)}}[cov(\theta_{Ik}, \theta_{Ij}|\vec{\theta}_{pa(I)})] = E_{\vec{\theta}_{pa(i)}}\Bigg(\theta_{pa(ij)}(\frac{f_{ik}+1}{f_{ik}+g_{ik}} \prod_{w=1}^{k-1} \frac{g_{iw}+1}{f_{iw}+g_{iw}} + 1 - \theta_{pa(ik)})\Bigg) \tag{2.30}$$

As it was shown the condition in equation 2.19 guarantees the preservation of the hierarchical structure. However, as it is shown (See Appendix b) a linear relationship between $\vec{f}_I$ and $\vec{\theta}_I$ results in the reduction of the generalized Dirichlet distribution to the Dirichlet distribution. In order to retain the more general characters of Dirichlet distribution we propose using other functional relationships between $\vec{f}_I$ and $\vec{\theta}_{pa(I)}$. The nature of the chosen functions allows us to have more control over the reflection of the parent nodes over their children, as an example by choosing a quadratic relationship between $\vec{f}_I$ and $\vec{\theta}_{pa(I)}$ and considering the fact that the elements of $\vec{\theta}_{pa(I)}$ are fractions of one, we reduce the reflection character by choosing a quadratic form, whilst on the other hand by choosing a square root relationship between $\vec{f}_I$ and $\vec{\theta}_{pa(I)}$ we end up with a tighter relationship between the parent and the children nodes. Further increase in the parent-children node relationship is theoretically possible by increasing the order of the roots, however the increasing of the order likewise increases the model complexity and its sensitivity. In this work we focus on the square root relationship.

One problem that arises with a nonlinear choice of parent and child nodes relationship is that unlike the hierarchical Dirichlet distribution, it is not viable to find an exact relationship for equations 2.29 and 2.30. We propose a simplifying assumption, that is justifiable through experimental results. The proposition is as follows:

$$E[f_{Ij}(\theta_{pa(I)})] \approx f_{Ij}(E(\theta_{pa(I)})) \tag{2.31}$$

$$E[g_{Ij}] \approx \frac{(1 - \sum_{k=1}^{j}[E[\theta_{pa(I)}(k)]])}{E[\theta_{pa(I)}(j)]} E[f_{Ij}] \tag{2.32}$$

The derived equations in this subsection provide us with the necessary means for computing the LMMSE estimation for each of the nodes in the hierarchy. Unlike Dirichlet distribution, generalized Dirichlet distribution generated vectors that are not exchangeable [90]. This lack of exchangeability character prohibits us from using any further computation simplification as we did with the Dirichlet distribution. In order to derive a proper estimation, we thus begin with proper ini-tialization of the parameters. Next we update parameters estimates iteratively until a convergence criterion is met. In the following section we will show the results of applying our model on an image database.

2.2.1 Experimental results

In order to show the results of applying the method, we have performed a series of experiments on a dataset designed for object classification. We have also performed a success rate comparison with the other existing models in order to show the pros and cons of the model.

best suited for the classification of mono

Analysis of the Classification Capability of the Hierarchical Generalized Dirichlet Model

In order to make a fair comparison with the already existing models, a set of experiments has been done on the hierarchical generalized Dirichlet model to analyze both

its classification and categorization success rate in comparison with the hierarchical Dirichlet and the Naive Bayes models. In the following experiments we assume a square root relationship between the parent-children parameters, as explained previously. Like before we define the model success rate as the ratio between the total number of the correctly classified images in all classes against the total number of images. Figure 2-7 shows the recognition success rate of the model as a function of the number of visual words. The same as it was shown for the hierarchical Dirichlet

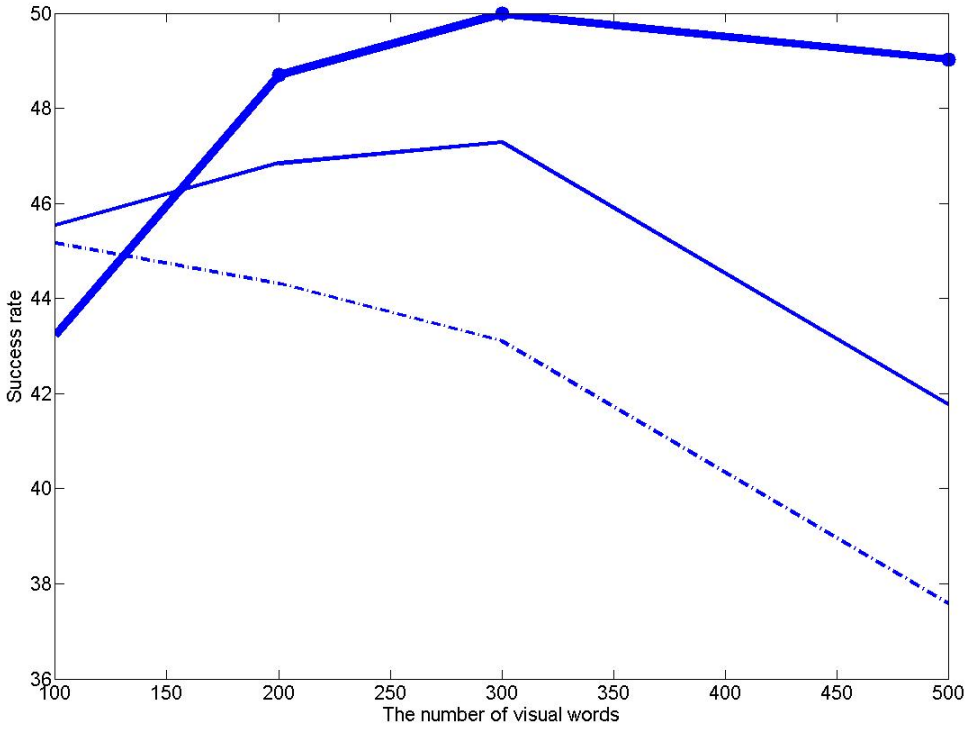


Figure 2-7: Comparison of the recognition success rate (In percent) of the hierarchical generalized Dirichlet model (Thick solid line) versus hierarchical Dirichlet model (Thin solid line) and the Naive Bayes model (Dashed line) for different numbers of visual words.

model in previous work [2], it is expected that the model shows the same considerable improvement by stepping higher in the hierarchy. Accordingly Figure 2-7 shows the second tier categorization success rate of the model versus the number of visual words. Table 2.2 shows the confusion matrix generated by optimal choice of system parameters. The same as the case for the hierarchical Dirichlet model it can be seen

that the existing errors are to a great extend contained within the boundaries of the hierarchy. This feature, as was shown in figures 2-7, leads to improved categorization rate of the model as we go upper inside the hierarchy. Same as it observed for the hierarchcial Dirichler model, increasing the number of visual words beyond the optimal point results in the model beginning to model clutter and noise and therefore. This in return leads to the drop of the total accuracy of the model.

Table 2.2: Optimal confusion matrix for the hierarchical generalized Dirichlet model.

Class	*apple*	*cow*	*cup*	*dog*	*horse*	*pear*	*tomato*
apple	54.9	2	15.7	4.3	2.6	4.3	26.3
cow	0	25.7	2.3	6.9	15.2	0.2	4.9
cup	4.3	1.7	61.4	8.6	2	1.4	7.7
dog	0.5	26	3.8	43.8	30.1	0.5	1.8
horse	0.5	34.3	5.2	39.8	38	0.2	4.6
pear	38.5	8	10.8	8.3	8.7	94.1	17
tomato	2	2	1.7	2.9	4.3	0	44.7

2.3 Hierarchical Beta-Liouville model

In the last two sections we developed two distinct hierarchical models based on the Dirichlet and generalized Dirichlet assumption. It was depicted that the generalized Dirichlet distribution in general offered better modeling capabilities in return for the need for making the model computationally complex. In order to make a trade off between the computational simplicity of the Dirichlet assumption and the efficiency of the generalized Dirichlet, in the following work we used a third prior assumption from the Liouville family of distributions that is the Beta-Liouville distribution. Having adapted our model for the Beta-Liouville assumption, we shall compare the three models to make a thorough analysis of the merits and drawbacks of each of the models in comparison with the other ones.

2.3.1 Beta-Liouville Distribution

We say that a vector $\vec{\theta} = \{\theta_1, ..., \theta_D\}$ follows a Liouville distribution with the parameters $\{\alpha_1, ..., \alpha_D\}$ and density generator $g(.)$ if the probability distribution function follows the following:

$$p(\vec{\theta}) = g(u) \times \prod_{d=1}^{D} \frac{\theta_d^{\alpha_d - 1}}{\Gamma(\alpha_d)} \tag{2.33}$$

In the above equation $u = \sum_{d=1}^{D} \theta_d < 1$ and $\theta_d > 0$. The mean, variance and the covariance of the vector are derived respectively as follows:

$$E(\theta_d) = E(u) \frac{\alpha_d}{\sum_{d=1}^{D} \alpha_d} \tag{2.34}$$

$$var(\theta_d) = E(u)^2 \frac{\alpha_d(\alpha_d + 1)}{(\sum_{m=1}^{D} \alpha_d)(\sum_{m=1}^{D} \alpha_m + 1)} - E(\theta_d)^2 \frac{\alpha_d^2}{(\sum_{m=1}^{D} \alpha_m)^2} \tag{2.35}$$

$$Cov(\theta_l, \theta_k) = \frac{\alpha_l \alpha_k}{\sum_{d=1}^{D} \alpha_d} \left(\frac{E(U^2)}{\sum_{d=1}^{D} \alpha_d + 1} - \frac{E(u)^2}{\sum_{d=1}^{D} \alpha_d} \right) \tag{2.36}$$

U is a random variable defined in the domain $U\,[0, 1]$ and it follows a density function $f(.)$ that is related to the density generator function as follows:

$$g(u) = \frac{\Gamma(\sum_{d=1}^{D} \alpha_d)}{u^{\sum_{d=1}^{D} \alpha_d - 1}} f(u) \tag{2.37}$$

As can be seen from equation 2.36, unlike Dirichlet distribution, the vectors are not necessarily negatively correlated. Therefore, the Liouville family of priors could offer a more realistic modeling in comparison to the Dirichlet distribution. One suitable distribution for $f(.)$ is the Beta distribution. Using Beta distribution with parameters (α, β) leads to the following PDF:

$$p(\vec{\theta}|\vec{\alpha}, \alpha, \beta) = \frac{\Gamma \sum_{d=1}^{D} \alpha_d) \Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \prod_{d=1}^{D} \frac{\theta_d^{\alpha_d - 1}}{\Gamma(\alpha_d)} \left(\sum_{d=1}^{D} \theta_d \right)^{\alpha - \sum_{d=1}^{D} \alpha_d} \left(1 - \sum_{d=1}^{D} \theta_d \right)^{\beta - 1} \tag{2.38}$$

In the above equation $\vec{\alpha} = \{\alpha_1, ..., \alpha_D\}$ and it gives the pdf function of the BL distribution.

For the Beta distribution with parameters (α, β), the mean and the second moment are as follows:

$$E(U) = \frac{\alpha}{\alpha + \beta} \tag{2.39}$$

$$E(U^2) = \frac{\alpha + 1}{\alpha + \beta + 1} E(U) \tag{2.40}$$

Combining the above with equation 2.36 derive the covariance matrix of the BL distribution. One of the desirable characters of the BL distribution is its conjugacy with the multinomial distribution. Therefore assuming that an observed vector $\vec{n} = (n_1, ..., n_D, n_{D+1})$ follows a multinomial distribution, which parameters follow a BL distribution with parameter space $(\vec{\alpha}, \alpha, \beta)$. The conjugacy between the BL and multinomial distribution leads to [16].

$$\theta|\vec{n} \sim BL(\vec{\alpha}', \alpha', \beta') \tag{2.41}$$

where $\sim BL$ indicates a vector generated by the multinomial Beta-Liouville distribution and in the above $\vec{\alpha'} = \vec{\alpha} + (n_1, ..., n_D)$, $\alpha' = \alpha + \sum_{d=1}^{d} n_d$ and $\beta' = \beta + n_{D+1}$. The above equations combined with equations 2.34, 2.35 and 2.36 leads to the derivation of the first and second order statistics of the BL distribution that we will use in our proposed model. Considering that $C = \left\{\vec{C_1}, ..., \vec{C_N}\right\}$ is the set of the count vectors that must be classified. The model assumes that the vectors are generated by a multinomial model with parameter space $\theta_I = \left\{\theta_{I1}, ..., \theta_{I(D+1)}\right\}$ as follows:

$$p(\vec{C_n}|\vec{\theta_I}) \propto \frac{(\sum_{d=1}^{D+1} C_{nd})!}{\prod_{d=1}^{K} C_{n(D+1)}!} \prod_{d=1}^{D+1} (\theta_{Id})^{C_{nd}} \tag{2.42}$$

In above $D + 1$ indicates the number of the elements inside $\vec{C_n}$ and c_{nd} indicates the dth element of $\vec{C_n}$.

In order to maintain the hierarchical structure the following condition must be met:

$$E[\vec{\theta_I}|\vec{\theta}_{pa(I)}] = \vec{\theta}_{pa(I)} \tag{2.43}$$

In above, $\theta_{pa(I)}$ indicates the generative parameter of the parent node of the $I-th$ node inside the hierarchy. Considering the above equations to hold, it was shown in [86] that by using the linear minimum mean square error (LMMSE) estimator one can find a general estimation of $\vec{\theta_I}$ parameter as follows.

$$\theta_i = E[\theta_i] + M^{-1} \times \begin{pmatrix} \hat{\theta}_{ch(i1)} - E[\theta_i] \\ \hat{\theta}_{ch(i2)} - E[\theta_i] \\ . \\ \hat{\theta}_{ch(im)} - E[\theta_i] \end{pmatrix} \tag{2.44}$$

where

$$M = \begin{pmatrix} \Sigma(\theta_{ch(i1)}) & \Sigma(\theta_{ch(i1)}\theta_{ch(i2)}) & \dots & \Sigma(\theta_{ch(i1)}, \theta_{ch(im)}) \\ \Sigma(\theta_{ch(i2)}, \theta_{ch(i1)}) & \Sigma(\theta_{ch(i2)}) & \dots & \Sigma(\theta_{ch(i2)}, \theta_{ch(im)}) \\ \vdots & \dots & \dots & \vdots \\ \Sigma(\theta_{ch(im)}\theta_{ch(i1)}) & \dots & \dots & \Sigma(\theta_{ch(im)}) \end{pmatrix}$$

In the above equation $\Sigma(\vec{\theta}_{ch(Ik)}, \vec{\theta}_{ch(Ij)})$ is the correlation matrix between the parameter vectors of the $k-th$ and $j-th$ children of the $I-th$ node assuming that the $I-th$ node has m child nodes inside the hierarchy.

It was shown in [86] that provided that the condition in equation 2.43 are met, the following simplifying relationships hold.

$$E[\vec{\theta}_{ch(I)}] = E[\vec{\theta_I}] \tag{2.45}$$

$$\Sigma(\vec{\theta_I}, \vec{\theta}_{ch(Ij)}) = \Sigma(\vec{\theta_I}) \tag{2.46}$$

$$\Sigma(\vec{\theta}_{ch(Ij)}, \vec{\theta}_{ch(Ik)}) = \Sigma(\vec{\theta_I}) \tag{2.47}$$

$$\Sigma(\vec{\theta}_{ch(Ik)}) = \Sigma\vec{\theta_I} + E_{\vec{\theta_I}}[\Sigma(\vec{\theta}_{ch(Ik)}|\vec{\theta_I})] \tag{2.48}$$

Hierarchical Beta-Liouville Model

In order to preserve the condition in equation 2.43, we propose that we use the following parent-child relationship inside the hierarchy.

$$\vec{\theta_I} \sim BL(\sigma\vec{\theta}_{pa(I)}, \xi\sigma\sum_{d=1}^{D}\theta_{pa(Id)}, \xi\sigma(1-\sum_{d=1}^{D}\theta_{pa(Id)}))$$

The proof of the preservation of the hierarchical structure considering equation 2.34 is as follows.

$$E[\vec{\theta_I}|\vec{\theta}_{pa(I)}] = \frac{\xi\sigma\sum_{d=1}^{D}\theta_{pa(Id)}}{\xi\sigma\sum_{d=1}^{D}\theta_{pa(Id)} + \xi\sigma(1-\sum_{d=1}^{D}\theta_{pa(Id)})} \times \frac{\sigma\vec{\theta}_{pa(I)}}{\sigma\sum_{d-1}^{D}\theta_{pa(Id)}} = \vec{\theta}_{pa(I)} (2.49)$$

In the above equation ξ is a scalar value that defines the shape of the BL distribution. It is noteworthy to mention that setting the value of ξ to unity reduces the BL distribution to the Dirichlet distribution as described in [16].

The training data for each of the hierarchy nodes is assumed to be conjured into a single meta count data of the form $\vec{n_I} = (n_{I1}, ..., n_{I(D+1)})$ and $N_I = \sum_{d=1}^{D+1} n_{Id}$ Considering the conjugacy between BL and multinomial distributions and with direct replacement we have:

$$E[\vec{\theta_i}|\vec{n_I}] = \frac{\alpha_I + \sum_{j=1}^{D} n_j}{\alpha_I + \beta_I + _I} \frac{\vec{\alpha}_I + \vec{n_I}}{\sum_{d=1}^{D}\alpha_{Id} + \sum_{d=1}^{D} n_{Id}} \quad (2.50)$$

where the parameters of BL are taken from equation 2.49. The next step is the derivation of the conditional covariance matrix of the hierarchical BL distribution. To that end it is necessary to derive the second moment of the posterior Beta distribution from equation 2.40. Assuming that $\sum_{d=1}^{D} n_{Id} >> 1$ by direct replacement we have:

$$E(U_I^2|\vec{n}_I) = (\frac{\alpha_i + \sum_{j=1}^{D} n_{Ij}}{\alpha_i + \beta_i + N_I})^2 \quad (2.51)$$

Inserting the above equation in equation 2.36, while considering equation 2.50 and simple algebraic simplifications give the following equation for the posterior covariance

between the elements of vector θ_I:

$$Cov(\theta_{Il}, \theta_{Ik}) \approx E[\theta_{Il}|\vec{n}_I]E[\theta_{Il}|\vec{n}_I] \times \frac{-1}{1 + \sigma \sum_{m=1}^{D} \theta_{pa(Im)} + \sum_{m=1}^{D} n_{Im}} \tag{2.52}$$

The posterior variance of the elements of $\vec{\theta_I}$ is likewise derived through inserting equations 2.51 and 2.50 into equation 2.35 and certain algebraic simplifications:

$$var(\theta_{Id}|\vec{n_I}) = E[\theta_{Id}|\vec{n_I}]^2[1 - \frac{E[\theta_{Id}|\vec{n_I}]^2}{(\alpha_I + \sum_{d=1}^{D} n_{Id})^2}] \tag{2.53}$$

The last two equations to be derived are the conditional expectation of the covariance and variance of the children nodes of the $I - th$ node. They're derived as follows accordingly:

$$E_{\vec{\theta_I}}[cov(\theta_{ch(Il)}, \theta_{ch(Ik)})|\vec{\theta_I}] \approx \frac{-1}{1 + \sigma \sum_{d=1}^{D} \theta_{Id}}[\theta_{Il}\theta_{Ik}] \tag{2.54}$$

$$E_{\vec{\theta_I}}[var(\theta_{ch(Il)})|\vec{\theta_I}] \approx \theta_{Il}^2[1 - \frac{\theta_{Il}^2}{(\sigma \sum_{d=1}^{D} \theta_{Id})^2}] \tag{2.55}$$

For the above approximations to hold it is necessary that for each of the nodes we have enough training data to satisfy the approximation condition. Through applying equations and approximations 2.51 to 2.55 in 2.45 to 2.48 a, we derive an iterating algorithm that estimates the value of $\vec{\theta_I}$ for all the hierarchy nodes.

Inference Algorithm

The inference algorithm is described as follows:

1. Assuming unity value for the parameters of the topmost node, generate a random initial vector for each of the nodes by Equation 2.49 Until a convergence criterion is met.

2. Estimate the posterior mean of the parameters from equation 2.50

3. Estimate the posterior covariance matrix from equations 2.53 and 2.52

4. Using the current estimation of $\vec{\theta}_I$ to extract the value of equation 2.54

5. Use equation 2.44 to extract the updated value of $\vec{\theta}_I$ for all nodes.

In this work the algorithm converges when the change in the estimated posterior mean remains below a threshold.

2.3.2 Experimental Results

In this section we compare our model with three other count data models. The Naive Bayes model [27] as the basis of count data modeling, the hierarchical Dirichlet model [2] and the hierarchical generalized Dirichlet model [2].

Analysis and Comparison of the Classification and Categorization Success Rate of the Model

We compared the classification success rate of our model in different tiers with the Naive Bayes [27], hierarchical Dirichlet [2] and hierarchical generalized Dirichlet models [3]. Figure 2-8 compares the model first tier classification success rate. As it can be seen from the figure 2-8.

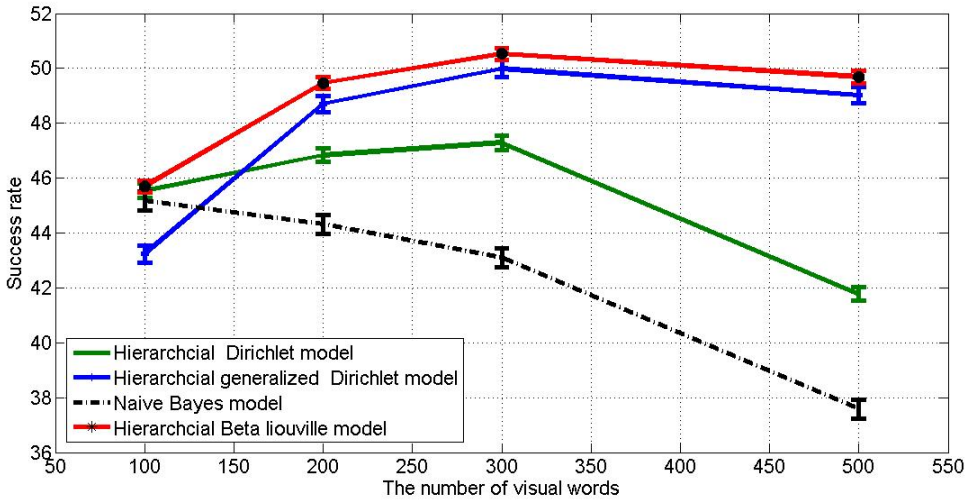


Figure 2-8: Comparison of the recognition success rate of the different models. The error bars are set at 90% standard deviation of the relative graphs.

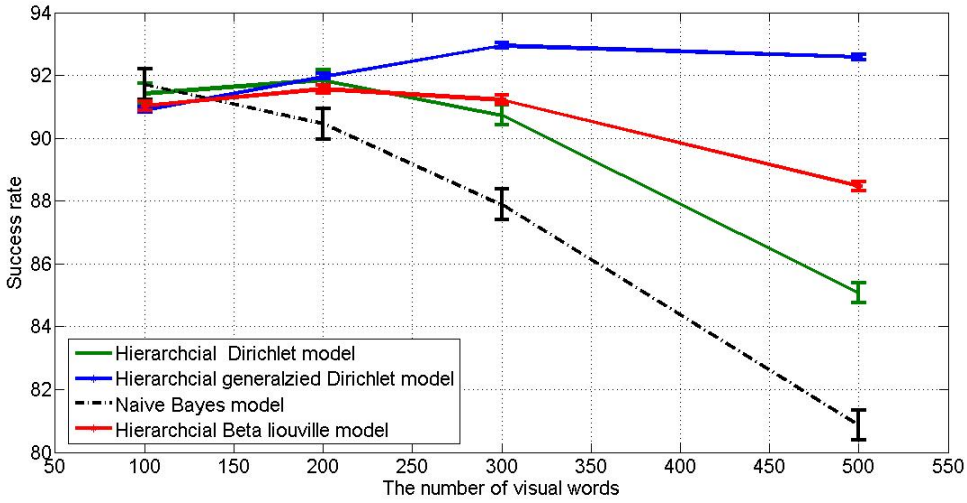


Figure 2-9: Comparison of the second tier recognition success rate of the different models. The error bars are set at 90% standard deviation of the relative graphs.

Figure 2-9 shows the second tier classification success rate of the different models. As it can be seen from the figure again the model shows superiority in comparison with Naive Bayes model and is in relative parity with hierarchical Dirichlet. Hierarchical generalized Dirichlet model on the other hand appears to outperform in upper tier classification. we suggest that this is mainly attributed to the fact that the generalized Dirichlet assumption offers a more versatile covariance matrix in comparison to the BL assumption. It must noted that the hierarchical generalized Dirichlet model requires solving the nonlinear square root equation as mentioned in [2] and its more sensitive due to twice the number of estimated parameters.

The confusion matrix table of our model is given in table 2.3. As it can be seen from this table, the hierarchical assumption has led to the classification error to be confined in the sibling nodes.

Table 2.3: Optimal confusion matrix for the hierarchical Beta-Liouville model.

Class	*apple*	*cow*	*cup*	*dog*	*horse*	*pear*	*tomato*
apple	49.1	0.8	15.6	3.4	2.3	6.6	16.7
cow	0	21.3	7.2	5.4	15.8	0	3.1
cup	6	0	52	3.7	0	1.7	1.1
dog	0	24.2	3.4	34.9	17.9	1.1	3.7
horse	0.5	50	9.5	48.8	58.9	0.5	12.4
pear	35	2.8	9.6	2.3	3.4	89.5	15
tomato	9	0.2	2.6	1.1	1.4	0.2	47

Chapter 3

Semisupervised Learning Hierarchical Structures

In the previous chapter we showed the merits of using hierarchical structures for improving the classification efficiency. The main problem with the proposed method was its requirement for the hierarchical structure to be known in advance. Even though for small datasets the assumption poses little problem, it is a restraining factor when dealing with huge datasets with numerous branches and layers. This constrain led us to devising the development of hierarchical models capable of stemming the hierarchy from primary crude guesses. The outcome of our work resulted in three distinct semisupervised learning hierarchy models based on the Dirichlet, Generalized Dirichlet and Beta-Liouville prior assumptions.

One of the main challenges in hierarchical object classification is the derivation of the correct hierarchical structure. The classic way around the problem is assuming prior knowledge about the hierarchical structure itself. Two major drawbacks result from the former assumption. Firstly it has been shown that the hierarchies tend to reduce the differences between adjacent nodes. It has been observed that this trait of hierarchical models results in a less accurate classification. Secondly the mere assumption of prior knowledge about the form of the hierarchy requires an extra amount of information about the dataset that in many real world scenarios may not be available. In this chapter we address the mentioned problems by introducing

online learning of hierarchical models. Our models start from a crude guess of the hierarchy and proceed to figure out the detailed version progressively. We show the merits of the proposed work via extensive simulations and experiments on a real objects database. The basic hierarchical model that we have used for developing our hierarchical semisupervised online learning approach. The model was originally proposed as a special case of the Dirichlet prior used in [86].

3.1 The Model

Looking back at Table 2.3 gives us an overview of the problem that needs to be dealt with. If one looks, for example, at the row showing the attributions to the class "horse" one observes that the class has a tendency to absorb a great portion of the objects which have visual similarity to it. We call it an absorbing class. The original model uses maximum likelihood (ML) method for classification. Therefore, it is logical to assume that the absorbing node tends to have the higher likelihood in comparison to the neighboring nodes. In order to improve the classification process, it is necessary that one finds a way for penalizing the absorbing ML. To achieve this end we proceed with defining a saliency factor for each node. One factor to be considered as a relatively reliable saliency factor is that similar objects in general give somehow the same number of visual words. The number of features extracted from an object follows a natural process, therefore it is expectable to assume that it can be modeled by normal distribution. The histogram of the number of features in each category is shown in figure 3-1. As the first step we redefine the likelihood of the count vector $\vec{C}_n$ to represent the $I-th$ class as follows:

$$p(\vec{C}_n|\vec{\theta}_I,\Theta(I)) \propto \frac{(\sum_{d=1}^{D+1} C_{nd})!}{\prod_{d=1}^{D+1} C_{n(d)}!} \prod_{d=1}^{D+1} (\theta_{Id})^{C_{nd}} \times p((\sum_{d=1}^{D+1} (\vec{C}_n)|\Theta(I))) \tag{3.1}$$

In above $\Theta(I)$ represents the statistical characteristics of the $I-th$ class. Therefore, assuming normal distribution for the number of feature occurrences in the $I-th$

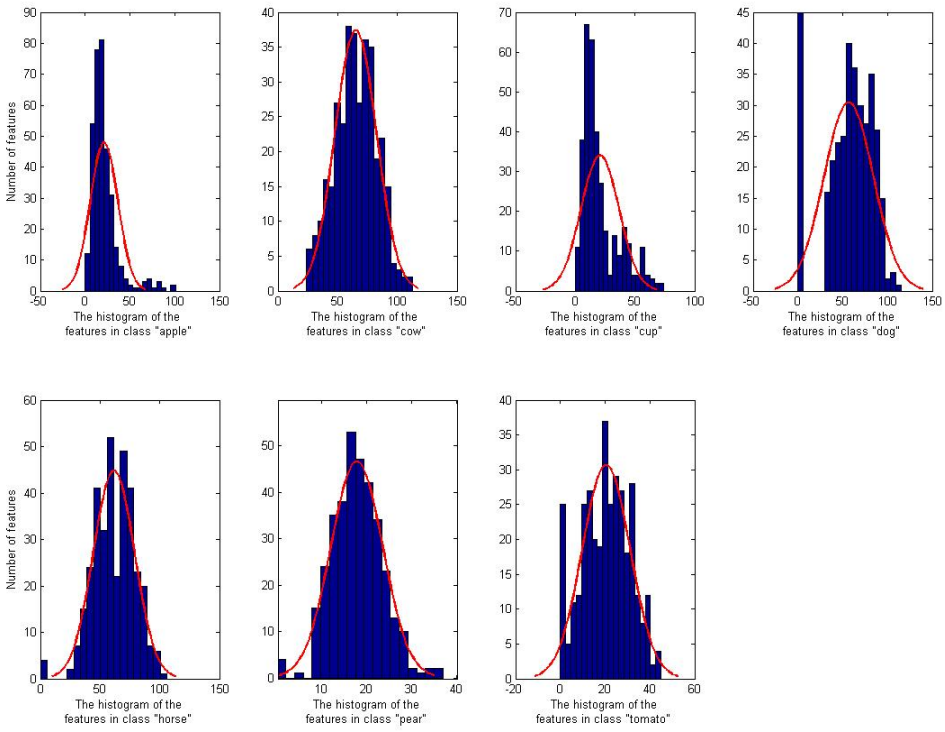


Figure 3-1: Histogram of the number of features present in each experimented class.

class, $\Theta(I)$ would be defined by the mean and the variance of the class histogram. It should be noted that $\Theta(I)$ is independent of $\vec{\theta_I}$ and therefore acts solely as a weighing factor, penalizing deviations from the established characters of the class.

There is yet another factor that needs to be considered for improving the model. As it was discussed in the previous section, in the original model we encounter dominant classes that tend to bias the classification process towards themselves. Mathematically the bias happens because $\vec{\theta_I}$ of the dominant class, offers a broad histogram of likelihood with comparably long tails. Therefore, theoretically there are always dominant classes present inside the model in the form of those classes which have stronger spreads on the log-likelihood spectrum. The model therefore finds out the hierarchical structure most efficiently when there is a strong similarity between the sibling classes and strong dissimilarity between the non related classes. In figure 3-2 we show the log-likelihood of the dominant classes in our experiments to further elaborate this fact.

Unlike the previous case, however, as it can be seen in figure 3-2, the log likelihood

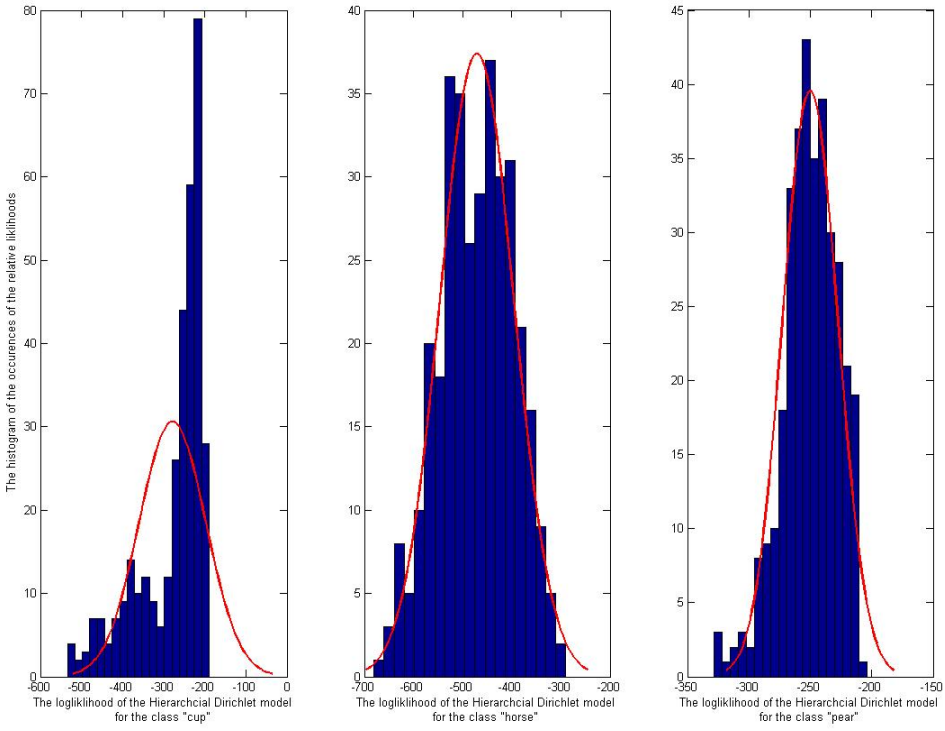


Figure 3-2: Log liklihood of the count data for the dominant classes.

of the count data does not clearly follow a Bell shaped Gaussian distribution and therefore it is analytically difficult to find a fitting function that covers all different shapes of the different log likelihoods. However, through observing the log likelihood of the dominant classes an effective boundary can be assumed where the majority of the likelihood instances occur. In our experiments we have observed that where normal fitting is possible the best results appear when the boundary is assumed to be one standard deviation from the mean of the training data likelihood. In theory the model accuracy suffers where the normal fitting fails to properly model the log likelihood. Still experiments show that the assumption is reliable in the majority of circumstances. By assigning this boundary on the dominant class, we devise an extra layer of protection against misplaced classification. As follows a new object is solely assigned to the dominant class when its likelihood falls in the acceptable boundary. If it doesn't, even if it shows a higher likelihood than other classes in the

same branch, it is rejected as an object belonging to the dominant class and the next highest likelihood is chosen as the assigned class.

Learning Hierarchical Structure

The presence of the dominant classes provides us with yet one more assumption easement. So far the hierarchical models proposed based on [86] have assumed a known hierarchical structure a priori. As an example the visual hierarchy used in [2] is brought in figure 3-3. In this work, however, we propose a learning hierarchical

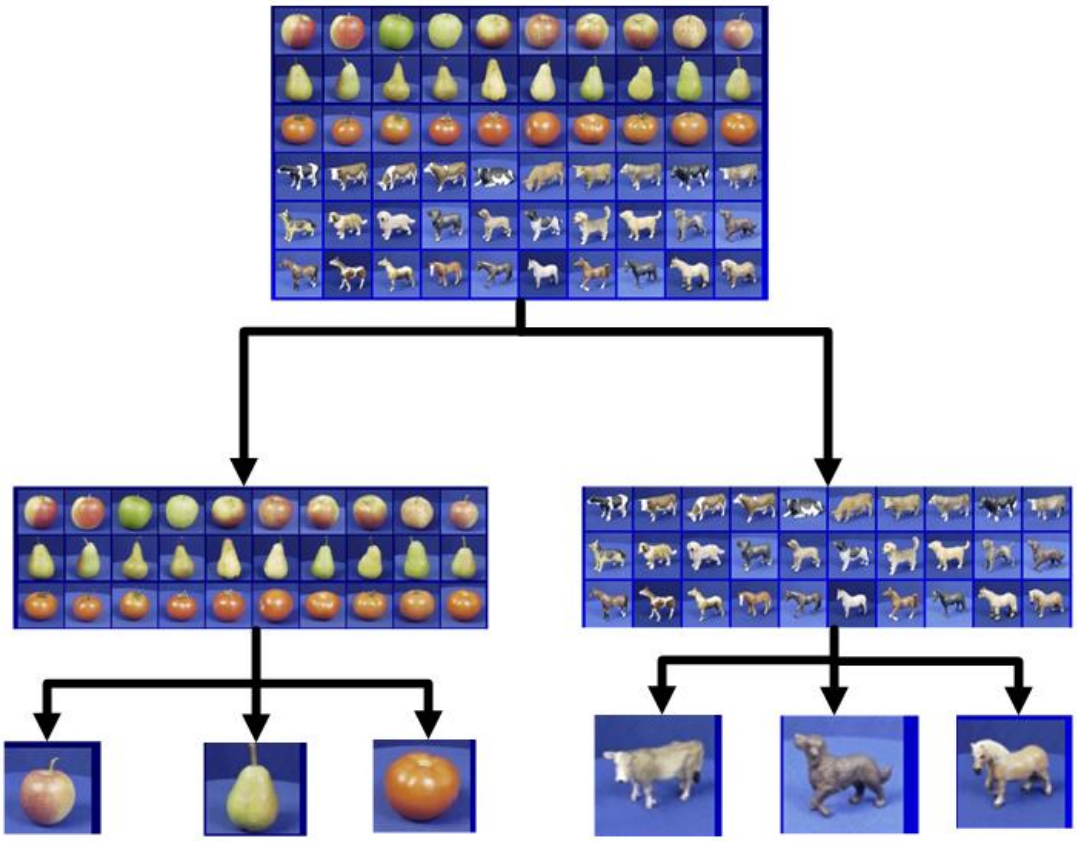


Figure 3-3: An example of hierarchical object classification.

structure based on the presence of dominant classes.We call our model semisupervised online learning of hierarchal structures (SOLHS). SOLHS starts from a crude sketch of the hierarchy, where only the dominant classes are placed in their relative positions inside the hierarchy. To derive the dominant classes we turn to the naive Bayes

classification over the training set. The dominant classes tend to give high likelihood not only to themselves but also to their sibling nodes, therefore they absorb the sibling entries in the confusion matrix. Theoretically there are always dominant classes, however the stronger the dominance of the class over its sibling classes gets the stronger the model efficiency in properly categorizing the data becomes. Here we need to define the difference between classification and categorization in our context. We define classification as the ability of the model to correctly identify different classes while we define categorization as the model ability to identify the concept the class belongs to. As an example a strong classifier can strongly tell the difference between a horse and a cow, while a strong categorizer can strongly depict that a horse or a cow belong to the animal class. As we described in this section the likelihood of the classes typically falls within a derivable boundary. Assuming that a new class appears which likelihood does not fall in the acceptable boundary of any of the dominant classes; SOLHS will decide that a new class of objects has been introduced. However, it is expectable to assume that the new class will have likelihood boundaries near to that of one of the dominant classes. In this step SOLHS compares the likelihood of the object with different dominant classes and decides where exactly the new branch of the hierarchy the new class must be placed. In this work, the assumption is that the new objects arrive in unlabeled classes. Totally random data arrival requires count data clustering as mentioned in the following works [66, 65]. SOLHS waits until enough new objects have arrived to form an appropriate training set. In the next step it assumes a new branch added to the hierarchy and it recalculates the model parameters [2, 3, 4] while including the new class. The process continues in the presence of coming data. Every time SOLHS decides that a new class has to be formed it adds the appropriate branch and recalculates the parameters accordingly. The following steps define the semisupervised online learning of the hierarchical structure phase:

1. From the training dataset extract the dominant classes.

2. From the training dataset extract the saliency and log likelihood of the dominant classes.

3. For each new entry find the nearest dominant class based on the maximum likelihood.

4. If the new entry does not fit within the salient boundaries of the dominant class flag the entry as belonging to an unidentified sibling node of the dominant class and repeat the process.

5. Once enough entries for the unidentified nodes is collected, re-estimate the model parameters with the inclusion of the unidentified nodes.

For the classification part we follow the following steps:

1. Use the ML estimation and if the ML remains within the salient boundaries of the dominant node with the highest ML select the dominant node as the class.

2. If the saliency fails enter the learning mode and perform the learning algorithm step 3-5.

Different Considered Priors

In this work, we analyzed three different prior distributions to be used for our model. The three distributions are: Dirichlet distribution, generalized Dirichlet distribution and Beta-Liouville distribution. By appropriate considerations, the three distributions satisfy the conditions in 2.43. Also the three distributions are known to be conjugate priors to the multinomial distribution, which is the second necessary condition for creating the hierarchical structure of [86].
A random vector $\vec{\theta_i}$ follows a Dirichlet distribution with parameter vector $\vec{\alpha_i} = (\alpha_{i1}, \ldots, \alpha_{i(D+1)})$ over the hyper plane $\sum_{k=1}^{D+1} \theta_{ik} = 1$, if its joint probability density function (PDF) is defined as follows [67]:

$$p(\vec{\theta_i}|\vec{\alpha_i}) = \frac{\prod_{k=1}^{D+1} \Gamma(\alpha_{ik})}{\Gamma(\sum_{k=1}^{D+1} \alpha_{ik})} \prod_{k=1}^{D+1} \theta_{ik}^{(\alpha_{ik}-1)} \tag{3.2}$$

where Γ is the Gamma function. Dirichlet distribution satisfies condition 2.43 unconditionally. Assuming $\vec{n_i} = (n_{i1}, \ldots, n_{i(D+1)})$ to be the observed vector, the conjugacy

with the multinomial distribution is derived as follows:

$$p(\vec{\theta}_i|\vec{n}_i) \propto \mathcal{D}(\alpha_{i1} + n_{i1}, \ldots, \alpha_{i(D+1)} + n_{i(D+1)}) \tag{3.3}$$

Defining $\vec{\alpha}_i^{'} = \vec{\alpha}_i + \vec{n}_i$, we obtain [39]:

$$E(\vec{\theta}_i|\vec{n}_i) = \frac{\vec{\alpha}_i^{'}}{|\vec{\alpha}_i^{'}|} \tag{3.4}$$

The second distribution that we use in our model is the generalized Dirichlet distribution. Following the same terminology used for Dirichlet distribution a random vector $\vec{\theta}_i$ defined over the hyper plane $\sum_{k=1}^{D} \theta_{ik} < 1$ is said to follow a generalized Dirichlet distribution with parameter space $\vec{\xi}_i = (\alpha_{i1}, \ldots, \alpha_{iD}, \beta_{i1}, \ldots, \beta_{iD})$, if its joint PDF is as follows:

$$p(\vec{\theta}_i|\vec{\xi}_i) = \prod_{k=1}^{D} \frac{\Gamma(\alpha_{ik} + \beta_{ik})}{\Gamma(\alpha_{ik})\Gamma(\beta_{ik})} \theta_{ik}^{\alpha_{ik}-1} (1 - \sum_{j=1}^{k} \theta_{ij})^{\gamma_{ik}} \tag{3.5}$$

Generalized Dirichlet distribution is also a conjugate prior to multinomial distribution and for $\vec{\theta}_i|\vec{n}_i \propto GD(\alpha_{i1}^{'}, \ldots, \alpha_{iD}^{'}, \beta_{i1}^{'}, \ldots, \beta_{iD}^{'})$, where:

$$\alpha_{ik}^{'} = \alpha_{ik} + n_{ik} \tag{3.6}$$

$$\beta_{ik}^{'} = \beta_{ik} + \sum_{l=k+1}^{D+1} n_{il} \tag{3.7}$$

and therefore we have [24]:

$$E(\theta_{ik}|\vec{n}_i) = \frac{\alpha_{ik}^{'}}{\alpha_{ik}^{'} + \beta_{ik}^{'}} \prod_{j=1}^{k-1} \frac{\beta_{ij}^{'}}{\alpha_{ij}^{'} + \beta_{ij}^{'}} \tag{3.8}$$

The following derivations provide the necessary conditions for maintaining the hierarchy:

$$\vec{\theta}_i \sim \begin{cases} \mathcal{GD}(\eta, \ldots, \eta, \zeta, \ldots, \zeta) & \text{if } i \text{ is the first node} \\ \mathcal{GD}((f(\vec{\theta}_{pa(i)}), g(\vec{\theta}_{pa(i)}))) & \text{otherwise} \end{cases} \tag{3.9}$$

where $\vec{\theta}_{pa(i)}$ indicates the parent of the $i-th$ node. The functions $f(\vec{\theta}_{pa(i)})$ and $g(\vec{\theta}_{pa(i)})$ depend on the parent node and must be determined in the way that the condition in Eq. 2.43 holds. By defining $\vec{f_i}$ and $\vec{g}_i$ functions as $\vec{f_i} = \{f_{i1}, \ldots, f_{iD}\}$ and $\vec{g}_i = \{g_{i1}, \ldots, g_{iD}\}$, it was shown in [3] that the following condition preserves the hierarchical structure:

$$g_i(k) = \frac{(1 - \sum_{l=1}^{k} \theta_{pa(i)}(l))}{\theta_{pa(i)}(k)} f_i(k) \tag{3.10}$$

It was shown in [3] that by choosing a linear relationship between $\vec{f_I}$ and $\vec{\theta}_{pa(I)}$ the hierarchical generalized Dirichlet model is reduced to a simple hierarchical Dirichlet model. It was thus suggested that a nonlinear relationship between $\vec{f_I}$ and $\vec{\theta}_{pa(I)}$ should be considered. Based on that assumption a square relationship between the parameters is considered as follows:

$$f_i(k) \propto (\theta_{pa(i)}(k))^2 \tag{3.11}$$

The reason behind the quadratic choice of dependency is that the choice allows a stricter relationship between $\theta_{pa(i)}$ and θ_i The last prior that we consider for our model is the Beta-Liouville distribution. A random vector $\vec{\theta_i}$ defined over the hyper plane $\sum_{k=1}^{D} \theta_{ik} < 1$ is said to follow a Beta-Liouville distribution with parameter space $(\{\alpha_1, \ldots, \alpha_D\}, \alpha, \beta)$, if its joint PDF is as follows:

$$p(\vec{\theta_i}|\vec{\alpha}, \alpha, \beta) = \frac{\Gamma \sum_{d=1}^{D} \alpha_d) \Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \prod_{d=1}^{D} \frac{\theta_{id}^{\alpha_d - 1}}{\Gamma(\alpha_d)} \left(\sum_{d=1}^{D} \theta_{id} \right)^{\alpha - \sum_{d=1}^{D} \alpha_d} \left(1 - \sum_{d=1}^{D} \theta_{id} \right)^{\beta - 1} \tag{3.12}$$

The condition for preserving the hierarchical structure with Beta-Liouville assumption was derived in [4] and is as follows:

$$\vec{\theta_i} \sim BL(\sigma \vec{\theta}_{pa(i)}, \xi\sigma \sum_{d=1}^{D} \theta_{pa(id)}, \xi\sigma(1 - \sum_{d=1}^{D} \theta_{pa(id)}))$$

Beta-Liouville distribution is also a conjugate prior of the multinomial distribution and we have:

$$\theta|(\vec{n}_i, \vec{\alpha}, \alpha, \beta) \sim BL(\vec{\alpha}', \alpha', \beta') \tag{3.13}$$

where $\sim BL$ indicates a vector generated by the Beta-Liouville distribution and in the above $\vec{\alpha'} = \vec{\alpha} + (n_{i1}, \ldots, n_{iD})$, $\alpha' = \alpha + \sum_{d=1}^{D} n_{id}$ and $\beta' = \beta + n_{iD+1}$. And we therefore have [48]:

$$E[\vec{\theta}_i|\vec{n}_i] = \frac{\alpha + \sum_{j=1}^{D} n_{ij}}{\alpha + \beta + |\vec{n}_i|} \frac{\vec{\alpha}_i + \vec{n}_i}{\sum_{d=1}^{D} \alpha_d + \sum_{d=1}^{D} n_{id}} \tag{3.14}$$

In the next section we show the results of applying SOLHS with three different prior assumptions and we compare its performances against the previously derived models.

3.2 Experimental Results

Image Dataset

To maintain consistency with previous works [2, 3, 4], we have chosen the ETH-80 dataset [51] for our experiments. The dataset is optimized for object classification purposes. It contains views and segmentation masks of 80 objects, each one photographed in more than 40 different poses. In total it contains more than 3000 images. There are 8 object classes, from which we choose 7 categories to validate our work. The choice of classes is based on visual similarities. In general 6 of them can be classified in 2 unique categories: fruits and animals. It was shown in previous works that the visual similarities between the chosen classes contribute much to the efficiency of the hierarchical classification. Approximately 20 percent of the image database is randomly chosen as the training set, whilst the remaining images form the test dataset.

Feature Extraction and Visual Words Generation

We use scale invariant feature transform (SIFT) descriptors [53] to represent our objects. The high dimensionality of the SIFT descriptors and its comparably robustness towards changes in scaling, illumination, occlusion, etc, compared with other feature descriptors, have been shown to result in better classification results [55]. To generate the visual vocabulary, we extract SIFT descriptors over the entire dataset. Each SIFT descriptor has a dimension of 128 as described in [53]. In the next step the K-Means algorithm [39] is used to extract the centroides and then construct the visual vocabulary that we shall use.

Hierarchical Model Generation

In previous works the optimal hierarchical structure for the dataset was shown to be the one displayed in figure 2-2. In SOLHS, however, our assumption is that only a crude structure of the hierarchy in figure 2-2 is known and the model proceeds with learning the rest of the structure as described in the previous section. The class "cup" acts as a misplaced class to show the effect of class misplacement in the system accuracy. We analyze and compare the strengths and the weaknesses of the model in classification and categorization in comparison with static models proposed in previous works. It will be shown in the following subsection that the current model offers a more efficient classification rate in expense of slightly decreasing the categorization efficiency in comparison with the static hierarchical models.

Analysis of the Recognition Capability of the Model

For recognition purposes the lowest branches of the hierarchy that show the individual object classes are analyzed. The main factor that affects the accuracy of the model is the number of chosen visual words. Each of the distributions has its own parent-children parameters that are extensively analyzed in previous works. In order to maintain the consistency we proceed with comparing the optimum results for each model against each other. The model recognition success rate is defined as the ratio

between the total number of correctly classified images in all classes against the total number of images. Figure 3-4 compares the recognition success rates of the different models as a function of the number of visual words. As it can be seen from this figure SOLHS for all distributions show better classification accuracy in comparison with its static counterpart. This is mostly due to the fact that through applying the online learning algorithm we have created a deeper distance between the sibling nodes and therefore we have improved the classification accuracy.

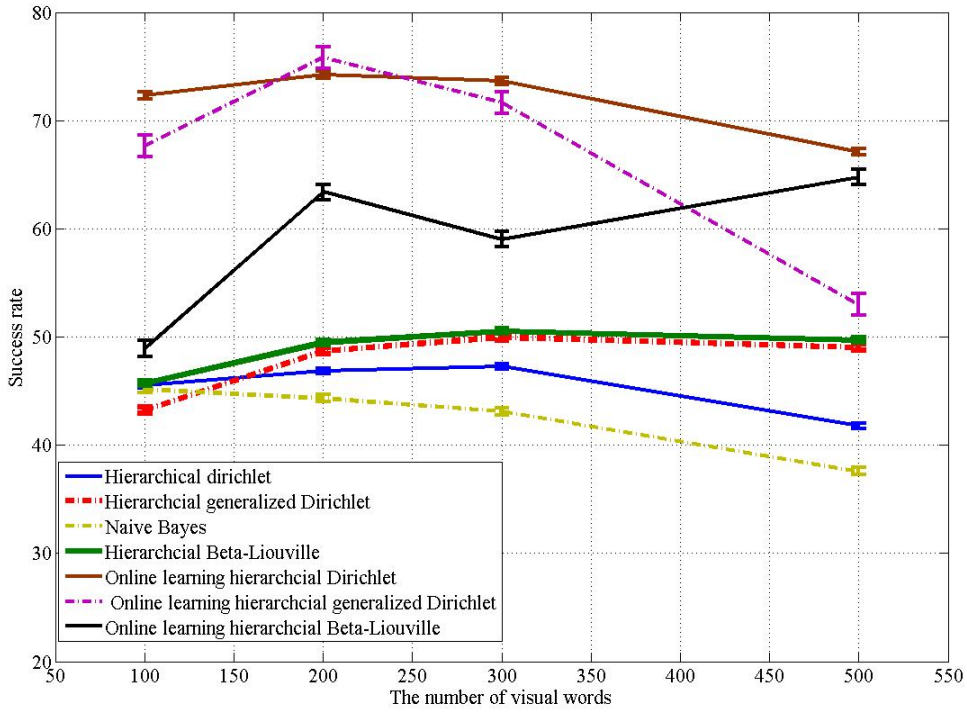


Figure 3-4: Comparison of the recognition success rates of SOLHS against the static models for different prior assumptions. The error bars are set at 90% standard deviation of the relative graphs.

Figure 3-5 shows the second tier categorization accuracy of SOLHS in comparison with the static hierarchical models. As it can be seen from this figure SOLHS in general acts less accurately when dealing with categorization task. The main reason behind the degradation of the categorization accuracy is due to the fact that the model starts from a crude understanding of the hierarchical structure. The static hierarchical models have the advantage of knowing in advance the parameters for the

entire nodes inside the hierarchy. On the other hand the learning model is prone to placement errors while it learns the correct structure. Since we assume that an object is classified mistakenly once will not be classified again we thus end up with higher misplacement errors in comparison to static models. This is further visualized by looking at the relative confusion matrices in tables 3.1, 3.2 and 3.3. As it can

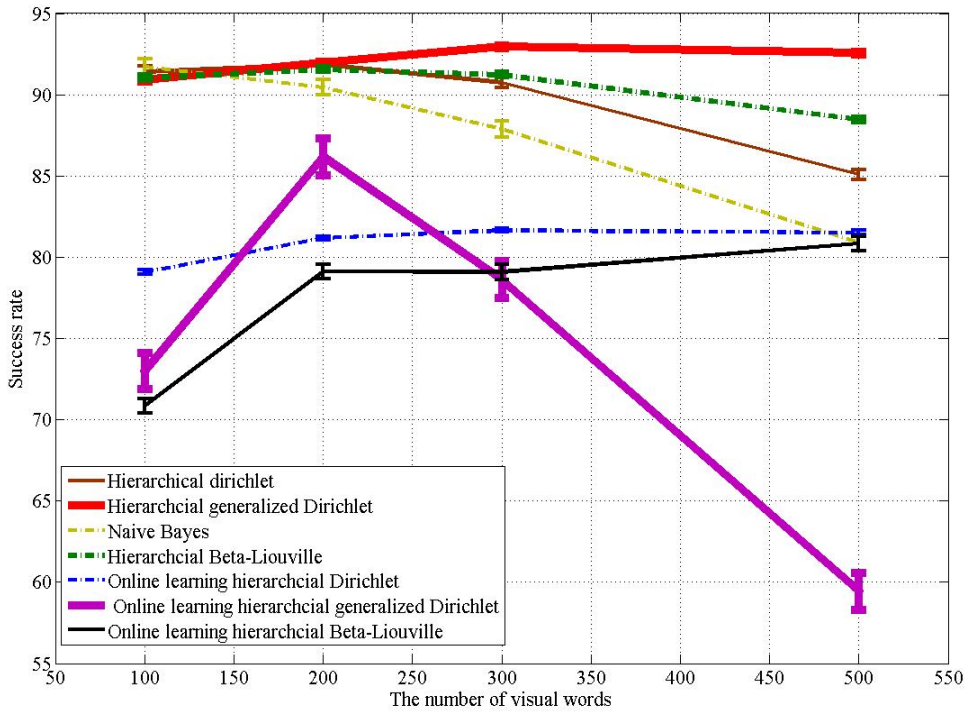


Figure 3-5: Comparison of the categorization success rates of the SOLHS against the static models for different prior assumptions. The error bars are set at 90% standard deviation of the relative graphs.

be seen from tables 3.1, 3.2 and 3.3 SOLHS progressively improves its performance through learning the hierarchical structure.

3.3 Conclusion

In this chapter we proposed a new adaptable general learning hierarchical model (SOLHS) dedicated to count data. As it was shown in the experimental results, SOLHS allows substantial improvement in hierarchical classification accuracy as com-

Table 3.1: Optimal confusion matrix of SOLHS when considering the online hierarchical generalized Dirichlet model.

Class	*cup*	*horse*	*pear*	*dog*	*cow*	*tomato*	*apple*
cup	231	13	8	7	1	33	37
horse	40	248	11	90	111	150	42
pear	71	81	323	9	11	41	61
dog	0	0	0	195	0	0	0
cow	0	0	0	0	223	0	0
tomato	0	0	0	0	0	98	0
apple	0	0	0	0	0	0	285

Table 3.2: Optimal confusion matrix of SOLHS when considering the online hierarchical Dirichlet model.

Class	*cup*	*horse*	*pear*	*dog*	*cow*	*tomato*	*apple*
cup	243	27	6	72	21	82	44
horse	23	232	0	0	0	58	1
pear	76	83	336	56	67	0	0
dog	0	0	0	173	0	0	0
cow	0	0	0	0	258	0	0
tomato	0	0	0	0	0	182	0
apple	0	0	0	0	0	0	297

Table 3.3: Optimal confusion matrix of SOLHS when considering the online Beta-Liouville model.

Class	*cup*	*horse*	*pear*	*dog*	*cow*	*tomato*	*apple*
cup	145	6	11	3	4	30	51
horse	123	303	12	91	110	129	13
pear	74	33	319	30	29	39	68
dog	0	0	0	177	0	0	0
cow	0	0	0	0	203	0	0
tomato	0	0	0	0	0	124	0
apple	0	0	0	0	0	0	210

pared to other models that we have described. The improvement is achieved through applying several saliency factors in SOLHS. In addition to that the learning algorithm proposed in SOLHS allows it to expand beyond the previously predefined hierarchical structures. SOLHS improves efficiency while dealing with unknown classes and as observed in the experiments succeeds in deciding the location of the new class within the hierarchy quite efficiently. SOLHS achieves this in return for a slight expense in

its categorization capability. Therefore, an interesting idea for further work on this model could be the design of learning models that reduce the misplacement of the data in the early learning phases.

Chapter 4

Variational Bayes Models for Count Data Classification

In this chapter we describe our contribution to multi-topic models. The models described in the previous chapters, though efficient in nature, dealt with single topic objects. While dealing with real life data most of the times it is necessary to be able to deal with data generated from multiple topics. As discussed in the introduction LDA model is known to be an efficient multi-topic generative model. However like other models based on the Dirichlet assumption the LDA model suffers from the Dirichlet assumption deficiencies. In this chapter in the beginning we introduce our adaptation of the LDA model using the generalized Dirichlet distribution as the model prior. We call this model Latent generalized Dirichlet allocation (LGDA). Afterwards we make a thorough comparison between the two models and show the merits and the drawbacks of our model in comparison to LDA. In order to extend the application framework we introduce natural scene classification and text classification applications to our framework as well. In the second section of this chapter we shall introduce our second multi-topic model based on the LDA model with the Beta-Liouville assumption instead. We call the second model latent Beta-Liouville allocation (LBLA) and we proceed with applying the model on different applications.

4.1 Latent generalized Dirichlet allocation

Count data appear in many domains (e.g. data mining, computer vision, machine learning, pattern recognition, bioinformatics, etc.) and applications. Examples include textual documents and images modeling and classification where each document or image can be represented by a vector of frequencies of words [70] or visual words [27], respectively. The extraction of knowledge hidden in count data is a crucial problem which has been the topic of extensive research in the past. The naive Bayes assumption, through the consideration of the multinomial distribution, was extensively used for count data modeling [70]. However, serious deficiencies were observed with the application of the multinomial distribution as thoroughly discussed in [54, 19]. The most widely used solution to overcome these deficiencies is the consideration of the Dirichlet distribution as a conjugate prior to the multinomial which generally offers better flexibility, generalization and modeling capabilities [54, 19]. Despite many favorable features, it has been pointed out that the Dirichlet distribution has some shortcomings, also. The main disadvantages of the Dirichlet distribution are its very restrictive negative covariance matrix and the fact that the elements with similar mean values must have absolutely the same variance which is not always the case in real-life applications [58]. To overcome those deficiencies, research has been focused on providing a transition from the Dirichlet assumption to better modeling assumptions [15]. The context of this chapter is majorly about this transition as well, where the ultimate goal is to have more accurate data modeling.

One of the immediate applications of proper data modeling is classification. It covers a vast extend of problems such as placement of textual data into appropriate library entries or classifying objects into their relevant categories. In this context, one of the most challenging tasks is the classification of natural scenes without going deep inside their semantics. The challenge behind the former is that natural scenes are generally composed of a huge number of minute objects. The presence of this ever occurring objects makes it extremely complicated to develop useful classifiers based on the semantics alone. After all one would expect to see roads, trees, sun and the

sky recurring in scenes both taken inside the city or in the suburb. The need to consider the presence of recurring data singletons, whether words, visual words or visual objects, led to the so-called topic based models. Latent semantic indexing (LSI) [28] is the first successful model proposed to extract recurring topics from data. It was proposed for textual documents modeling using mainly singular value decomposition (SVD). A generative successful extension of LSI called probabilistic latent semantic indexing (PLSI) was proposed in [42]. However, PLSI is only generative at the words layer and does not provide a probabilistic model at the level of documents. Therefore, two major problems arise with PLSI. Firstly, the number of parameters increases with the number of documents. Secondly, it is not clear how one can learn a document outside of the training phase. To overcome these shortcomings, the authors in [12] proposed the LDA model which has so far proven to be a reliable and versatile approach for data modeling. LDA has received a particular attention in the literature and several applications (e.g. natural scene classification [31]) and extensions have been proposed. Examples of extensions include the hierarchical version of LDA [11], used for instance in [79] for hierarchical object classification, and the online version proposed in [41]. Of course, these extension efforts are useful for several real-life applications and scenarios, but have ignored an important aspect of LDA namely the fact that it considers the Dirichlet distribution, and then its drawbacks, for generating latent topics.

Recently, it has been shown that generalized Dirichlet distribution is a good alternative to the Dirichlet when using finite mixture models for count data clustering [58]. Like the Dirichlet, the generalized Dirichlet distribution is a conjugate prior to the multinomial distribution which is crucial property in the LDA model. Moreover, the generalized Dirichlet has a more versatile covariance matrix and also it lifts the variance limitations facing Dirichlet vectors [58]. The goal of this work is to propose an extension of LDA based on the generalized Dirichlet distribution. Previously other researchers tried to develop latent topic models based on the conjugate priors other than Dirichlet [22]. Their model however is based on Gibbs sampling and Markov chain Monte Carlo (MCMC) method [23, 69]. The advantage of the MCMC method

is its relative ease of derivation. However it has been shown that sampling methods require much more computation time than the deterministic methods such as the variational Bayes. Therefore where it is possible to derive an analytic form, the deterministic models are more preferable. In this section we shall thus derive the extension to the LDA model using the generalized Dirichlet assumption using the variational Bayes method. To maintain consistency with the LDA model we call our model, latent generalized Dirichlet allocation (LGDA). We shall develop a variational Bayes estimation approach inspired from the one proposed in [12], yet with the generalized Dirichlet assumption. The Dirichlet distribution is a special case of the generalized Dirichlet distribution [24, 18], therefore it is expectable that the LGDA will provide good modeling capabilities. In the experimental results we shall elaborate the conjunctions between the two models further. We shall compare the two models via two challenging applications namely text and natural scene classification.

4.1.1 The Model

Like LDA, LGDA is a fully generative probabilistic model over a corpus. A corpus in our case is a collection of M documents (or images) denoted by $\mathbf{D} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_M\}$. Each document $\mathbf{w}_m$ is a sequence of N_m words $\mathbf{w}_m = (w_{m1}, \ldots, w_{mN_m})$. In what follows, for sheer convenience, we drop the index m wherever we are not referring to a specific document. The word w_n is a binary vector drawn from a vocabulary of V words, so that $w_n^j = 1$ if the $j-th$ word is chosen and zero otherwise. The model proceeds with generating every single word (or visual word) of the document (or the image) through the following steps:

1. Choose $N \propto Poisson(\zeta)$.
2. Choose $(\theta_1, \ldots, \theta_d) \propto GenDir(\vec{\xi})$.
3. For each of the N words w_n:
   (a) choose a topic $z_n \propto Multinomial(\vec{\theta})$.
   (b) Choose a word w_n from $p(w_n|z_n, \beta_w)$.

In above z_n is a $d+1$ dimensional binary vector of topics defined so that $z_n^i = 1$ if the $i-th$ topic is chosen and zero otherwise. We define, $\vec{\theta} = (\theta_1, \ldots, \theta_{d+1})$, where $\theta_{d+1} = 1 - \sum_{i=1}^{d} \theta_i$. A chosen topic is attributed to a multinomial prior β_w over the vocabulary of the words so that $\beta_{w_(ij)} = p(w^j = 1|z^i = 1)$, from which every word is randomly drawn. $p(w_n|z_n, \beta_w)$ is a multinomial probability conditioned on z_n and $GenDir(\vec{\xi})$ is a d-variate generalized Dirichlet distribution with parameters $\vec{\xi} = (\alpha_1, \beta_1, \ldots, \alpha_d, \beta_d)$ and probablity distribution function p:

$$p(\theta_1, \ldots, \theta_d|\vec{\xi}) = \prod_{i=1}^{d} \frac{\Gamma(\alpha_i + \beta_i)}{\Gamma(\alpha_i)\Gamma(\beta_i)} \theta_i^{\alpha_i - 1}(1 - \sum_{j=1}^{i} \theta_j)^{\gamma_i} \tag{4.1}$$

where $\gamma_i = \beta_i - \alpha_{i+1} - \beta_{i+1}$. It is straightforward to show that when $\beta_i = \alpha_{(i+1)} + \beta_{(i+1)}$, the generalized Dirichlet distribution is reduced to Dirichlet distribution [18]. With the above parameters, the mean and the variance matrix of the generalized Dirichlet elements are as follows [18]:

$$E(\theta_i) = \frac{\alpha_i}{\alpha_i + \beta_i} \prod_{k=1}^{i-1} \frac{\beta_k}{\alpha_k + \beta_k} \tag{4.2}$$

$$Var(\theta_i) = E(\theta_i)\left(\frac{\alpha_i + 1}{\alpha_i + \beta_i + 1} \prod_{k=1}^{i-1} \frac{\beta_k + 1}{\alpha_k + \beta_k} + 1 - E(\theta_i) \right) \tag{4.3}$$

and the covariance between θ_i and θ_j is given by:

$$Cov(\theta_i, \theta_j) = E(\theta_j) \times \left(\frac{\alpha_i}{\alpha_i + \beta_i + 1} \prod_{k=1}^{i-1} \frac{\beta_k + 1}{\alpha_k + \beta_k} + 1 - E(\theta_i) \right) \tag{4.4}$$

It can be seen from equation 4.4, that the covariance matrix of the generalized Dirichlet distribution is more general than the covariance matrix of the Dirichlet distribution and unlike Dirichlet distribution it is possible for two elements inside the random vector to be positively correlated. Also unlike Dirichlet two elements with the same mean value can have different variances. Generalized Dirichlet distribution, like the Dirich-

let distribution, belongs to the exponential family of distributions (see Appendix A.5). This means that the generalized Dirichlet distribution has a conjugate prior that can be developed in a formal way, which is an important property that we shall use in the following for the learning of our model. It turns out also that generalized Dirichlet like Dirichlet is the conjugate prior of the multinomial distribution. This implies that if $\vec{\theta}$ follows a generalized Dirichlet distribution with parameters $\vec{\xi}$ and $\vec{N} = (n_1, \ldots, n_{d+1})$, follows a multinomial with parameter $\vec{\theta}$, then the posterior distribution $p(\vec{\theta}|, \vec{\xi}, \vec{N})$ also follows a generalized Dirichlet distribution with parameters $\xi^{'}$ given as follows [58]:

$$\alpha_i^{'} = \alpha_i + n_i \tag{4.5}$$

$$\beta_i^{'} = \beta_i + \sum_{l=i+1}^{d+1} n_l \tag{4.6}$$

Having our generalized Dirichlet prior in hand, we proceed with defining the $(d+1)\times V$ word-topic probability matrix β_w which element $\beta_{w_{ij}} = p(w_j = 1|z_i = 1)$ shows the probability of drawing the $j-th$ word given that the $i-th$ latent topic is chosen. Like the LDA case, we proceed with assuming a non-generated β_w matrix, but we will show that this assumption does not have a serious impact and it can be revoked without bringing harm to the entire model. By assuming conditional independence of the variables, the same as LDA, one can deduce the following joint distribution:

$$p(\vec{\theta}, \mathbf{z}, \mathbf{w}|\vec{\xi}, \beta_w) = p(\vec{\theta}|\vec{\xi})p(\mathbf{w}|\mathbf{z}, \beta_w)p(\mathbf{z}|\vec{\theta}) \tag{4.7}$$

where $\boldsymbol{z}$ is the set of latent topics. Integrating over the $\vec{\theta}$ parameters and the topic space gives

$$\begin{aligned} p(\mathbf{w}|\vec{\xi}, \beta_w) &= \prod_{i=1}^{d} \frac{\Gamma(\alpha_i + \beta_i)}{\Gamma(\alpha_i)\Gamma(\beta_i)} \int \theta_i^{\alpha_i - 1}(1 - \sum_{j=1}^{i} \theta_j)^{\gamma_i} \\ &\times \prod_{n=1}^{N} \prod_{i=1}^{d+1} \prod_{j=1}^{V} (\theta_i \beta_{w_{ij}})^{w_n^j} d\vec{\theta} \end{aligned} \tag{4.8}$$

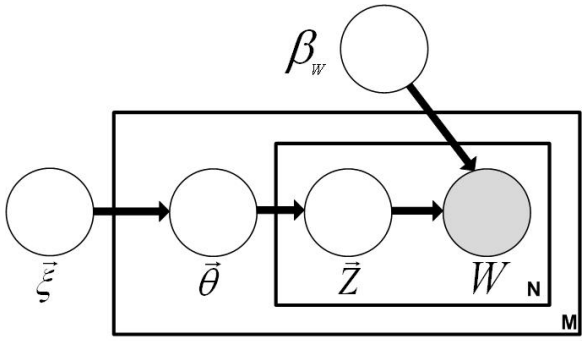


Figure 4-1: Graphical representation of LGDA model. The shaded circles show observed nodes. The blank circles are the hidden nodes. From outside to inside is the corpus space, the document space and the word space.

In the previous equation, $\vec{\xi}$ and β_w are the corpus level parameters that are selected once per each document in the corpus. $\vec{\theta}$ is the document level parameter and is chosen once per document. $\boldsymbol{z}$ and $\boldsymbol{w}$ are word level parameters and are chosen once per every word inside each document. Thus, we can obtain the probability of the corpus as follows:

$$p(\mathbf{D}|\vec{\xi}, \beta_w) = \prod_{m=1}^{M} p(\mathbf{w}_m|\vec{\xi}, \beta_w) \tag{4.9}$$

LGDA has basically the same probabilistic graphical model as LDA as it is shown in figure 4-1.

4.1.2 LGDA Inference

The main inference problem of LGDA is estimating the posterior of the hidden variables, $\vec{\theta}$ and $\boldsymbol{z}$:

$$p(\vec{\theta}, \boldsymbol{z}|\boldsymbol{w}, \vec{\xi}, \beta_w) = \frac{p(\vec{\theta}, \boldsymbol{z}, \boldsymbol{w}|\vec{\xi}, \beta_w)}{p(\boldsymbol{w}|\vec{\xi}, \beta_w)} \tag{4.10}$$

The above equation is known to be intractable. As proposed in [12], an efficient way to estimate the parameters in this intractable posterior is to use variational Bayes (VB) inference. VB inference offers a solution to the intractability problem by determining

a lower bound on the log likelihood of the observed data which is mainly based on considering a set of variational distributions on the hidden variables [46]:

$$q(\vec{\theta}, \boldsymbol{z}|\boldsymbol{w}, \vec{\xi_q}, \Phi_{\boldsymbol{w}}) = q(\vec{\theta}|\vec{\xi_q}) \prod_{n=1}^{N} q(z_n|\phi_n) \tag{4.11}$$

In the above $q(\vec{\theta}|\vec{\xi_q})$ can be viewed as a variational generalized Dirichlet distribution, calculated once per document, $q(z_n|\phi_n)$ is a multinomial distribution with parameter ϕ_n extracted once for every single word inside the document, and $\Phi_{\boldsymbol{w}} = \{\phi_1, \phi_2, \ldots, \phi_N\}$. Using Jensen's inequality [46] one can derive the following:

$$\log p(\boldsymbol{w}|\vec{\xi}, \beta_w) \geq E_q[\log p(\vec{\theta}, \boldsymbol{z}, \boldsymbol{w}|\vec{\xi}, \beta_w)] - E_q[\log q(\vec{\theta}, \boldsymbol{z})] \tag{4.12}$$

Assigning $L(\vec{\xi_q}, \Phi_{\boldsymbol{w}}; \vec{\xi}, \beta_w)$ to the right-hand side of the above equation it can be shown that the difference between the left-hand side and the right-hand side of the equation is the KL divergence between the variational posterior probability and the actual posterior probability, thus we have:

$$\log p(\boldsymbol{w}|\vec{\xi}, \beta_w) = L(\vec{\xi_q}, \Phi_{\boldsymbol{w}}; \vec{\xi}, \beta_w) + KL\big(q(\vec{\theta}, \boldsymbol{z}|\vec{\xi_q}, \Phi_{\boldsymbol{w}})||p(\vec{\theta}, \boldsymbol{z})|\boldsymbol{w}, \vec{\xi}, \beta_w)\big) \tag{4.13}$$

The left hand side of the above equation is constant in relation to variational parameters, therefore to minimize the KL divergence on the right-hand side one can proceed with maximizing $L(\vec{\xi_q}, \Phi_{\boldsymbol{w}}; \vec{\xi}, \beta_w)$. Up to here the formulation basically follows the LDA model. The divergence of the models begins when we proceed with assigning the generalized Dirichlet distribution as the parameter generator instead of the LDA Dirichlet assumption. In appendix A.6 we bring the breakdown of $L(\vec{\xi_q}, \Phi_{\boldsymbol{w}}; \vec{\xi}, \beta_w)$. Using variational inference to maximize the lower bound $L(\vec{\xi_q}, \Phi_{\boldsymbol{w}}; \vec{\xi}, \beta_w)$ with respect to ϕ_{nl}, we derive the following updating equations for the variational multinomial (see

Appendix A.6)

$$\phi_{nl} = \beta_{lv} e^{(\lambda_n - 1)} e^{(\Psi(\gamma_l) - \Psi(\gamma_l + \delta_l))} \tag{4.14}$$

$$\phi_{n(d+1)} = \beta_{(d+1)v} e^{(\lambda_n - 1)} e^{(\Psi(\delta_d) - \Psi(\gamma_d + \delta_d))} \tag{4.15}$$

whereΨ is the digamma function, $\beta_{lv} = p(w^v = 1 | z^l = 1)$ and the weighing constant $e^{\lambda_n - 1}$ is given by:

$$e^{\lambda_n - 1} = \frac{1}{\sum_{l=1}^{d} \beta_{lv} e^{(\Psi(\gamma_l) - \Psi(\gamma_l + \delta_l))} + \beta_{(d+1)v} e^{(\Psi(\delta_d) - \Psi(\gamma_d + \delta_d))}} \tag{4.16}$$

Maximizing the lower bound L with respect to the variational generalized Dirichlet parameter gives the following updating equations (see Appendix A.6):

$$\gamma_l = \alpha_l + \sum_{n=1}^{N} \phi_{nl}, \tag{4.17}$$

$$\delta_l = \beta_l + \sum_{n=1}^{N} \sum_{ll=l+1}^{d+1} \phi_{n(ll)}. \tag{4.18}$$

Comparing the above equations with equations 4.5 and 4.27 shows that the variational generalized Dirichlet for each document acts as a posterior in the presence of the variational multinomial parameters. The same conclusion was observed in [12] for the LDA case. This is a direct result of the conjugacy between the generalized Dirichlet and the multinomial distribution.

4.1.3 Parameter Estimation

The goal of this subsection is to find the model's parameters estimates based on the variational parameters derived in the last subsection. One needs to consider that the LGDA parameters are corpus parameters and therefore they are estimated by considering all M documents inside the corpus. In the following, we denote $L = \sum_{m=1}^{M} L_m$ as the lower bound corresponding to all the corpus, where L_m is the lower bound corresponding to each document m.

Maximizing the corpus lower bound L with respect to $\beta_{w(lj)}$ delivers the following updating equation (see Appendix A.6)

$$\beta_{w(lj)} \propto \sum_{d=1}^{M} \sum_{n=1}^{N_d} \phi_{dnl} w_{dn}^{j} \tag{4.19}$$

The model's parameters are the last ones to be derived. Following the work of Minka [56], it was shown in [12] that in order to derive LDA parameters it was feasible to use the Newton-Raphson algorithm for parameters estimation. It was also shown that due to the characteristics of the Dirichlet distribution, it is possible to exchange the computationally demanding problem of inverting the Hessian matrix of the Lower bound with a linear operator and therefore reducing the model complexity.

The Hessian matrix of the generalized Dirichlet distribution offers the same useful, albeit in a different way, simplification. This characteristic was analyzed in [18]. The nature of the generalized Dirichlet distribution leads the Hessian matrix to take a 2×2 block-diagonal shape. The inverse matrix of a block-diagonal matrix is another block-diagonal matrix consisting of the inverses of the blocks of the original matrix. Therefore the problem of inverting the $2d \times 2d$ Hessian matrix is reduced to computing the inverse of 2×2 matrix for d instances. The complete derivation of the model parameters is brought in Appendix 4.

The last formulation that we need to derive to prepare our model for the classification task is the likelihood of a document in our model. This can be done by deriving first the likelihood of a randomly chosen word w_n inside the document:

$$\begin{aligned} p(w_n|\vec{\xi}) &= \sum_{l=1}^{d+1} \int p(w_n|z_l) p(z_l|\vec{\theta}) p(\vec{\theta}|\vec{\xi}) d\vec{\theta} \\ &= \sum_{l=1}^{d+1} p(w_n|z_l) \int p(z_l|\vec{\theta}) p(\vec{\theta}|\vec{\xi}) d\vec{\theta} \\ &= \sum_{l=1}^{d} \beta_{l(v|w_n^v=1)} E[\theta_l] + \beta_{(d+1)(v|w_n^v=1)} (1 - \sum_{l=1}^{d} E[\theta_l]) \end{aligned} \tag{4.20}$$

Combining Eq. 4.2 with Eq. 4.20 delivers the formulation for the word likelihood as follows:

(4.21)

The log likelihood of a document $\boldsymbol{w}$ is derived as the sum of the log likelihoods of the words present inside the document and therefore we have:

$$logp(\boldsymbol{w}_m|\vec{\xi}) = \sum_{n=1}^{N_{\boldsymbol{m}}} \log p(w_n|\vec{\xi}) = \sum_{v=1}^{V} cnt_{mv} \log p(w_v|\vec{\xi}) \tag{4.22}$$

where for each document $\boldsymbol{w}$, cnt_v is the number of times the $v-th$ word is drawn.

4.1.4 Experimental Results

In this chapter we bring the results of applying the LGDA model on two distinct challenging applications namely text and natural scene classification. The former was introduced in [12] as the primary application of the LDA model. The later application was developed and adapted in [31] for LDA as well. In order to keep consistency with both works, we proceed with using the same datasets used in [12] and [31] to test our own model.

4.1.5 Text Classification

In text classification, the problem at hand is deciding which distinctive class to assign a given document to [78]. This problem can be looked upon from two distinct but related ways. Assuming that the number of the classes is known, from a first perspective text classification can be viewed as a binary categorization problem where the main problem is to decide to which class we should assign the text given two distinctively chosen classes. The other way to look upon the problem is to decide how accurately can the model assign the proper class to a document in the presence of all other classes. We proceed with giving results for each of the two mentioned scenarios in the following.

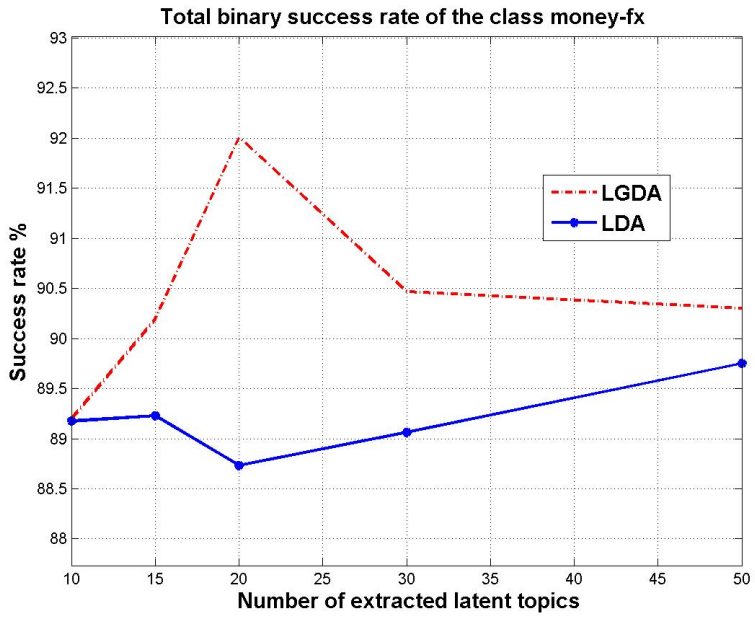
For our simulations, we chose the Reuters-21578 dataset [1]. This dataset consists of 21578 documents and in total there are more than 20000 words present inside it. Independent works have already classified most of the 21578 documents into superseding categories. Even though there are many extracted categories thus obtained, not all of them contain enough documents to be suitable for training and testing purposes. Thus, we limit ourselves to the top 6 categories extracted from the dataset. They, in total, comprise more than 9000 documents of the original dataset and nearly all the words present in the unabridged dataset. Table 4.1 describes the considered classes.

Class name	number of documents
'acq'	2293
'crude'	579
'earn'	3939
'grain'	593
'interest'	479
'money-fx'	729

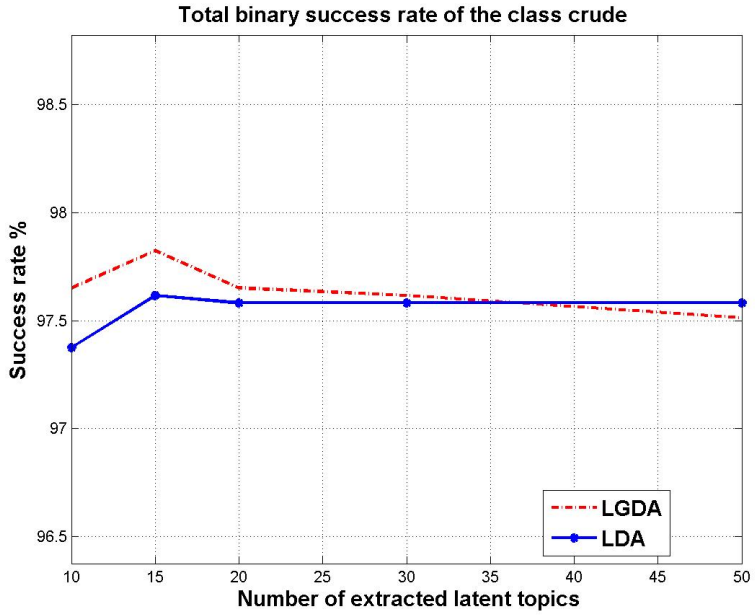
Table 4.1: Extracted classes and number of available documents per each class.

To examine the classification accuracy of the models, in the first step we choose a certain number of the documents in each of the classes as training documents. Next, we learn our models for each of the chosen training sets, for different numbers of latent topics to observe the effect of choosing them on the classification accuracy. Classification in this first experiment is regarded as a binary process meaning that the document is presented to two different trained classes and the class that gives the higher likelihood is chosen as the document class. Selected success rates of the two models under same training conditions are brought in figure 4-2. An interesting observation regarding the two models can be deduced from this figure. The Reuters dataset consists of relatively short documents that are presented as extremely sparse count vectors over the entire vocabulary set. Both LDA and LGDA use variational Bayes inference as their core learning method, however the sparsity of the count vectors causes both models to basically provide the same fit over the training set. The result is that facing classification vectors, the two models roughly offer the same suc-

[1] http://www.daviddlewis.com/resources/testcollections/reuters21578/



[a]



[c]

Figure 4-2: Comparison of binary classification success rate of the two models. Red line: LGDA, blue Line: LDA

cess rate. This can be seen in figure 4-2. There is an exception to this observation. When the two classes are inheritingly similar to each other, one may expect that the models fail to separate them as precisely as when the classes are mutually irrelevant. In these instances the model that offers the better fitting to its training set could

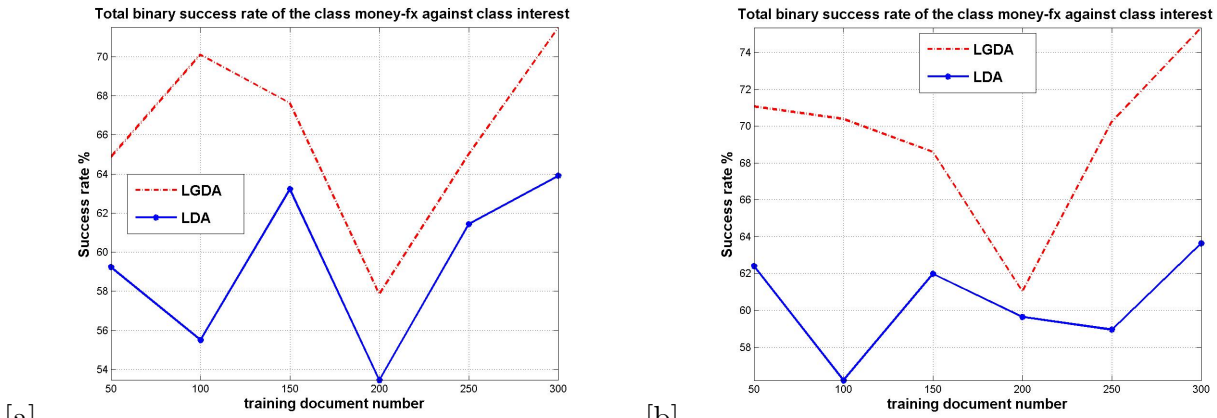


Figure 4-3: Comparison of binary classification success rate of the models for 'Money-fx' class for [a] 15 extracted latent topics [b] 30 extracted latent topics. Red line: LGDA, blue Line: LDA

offer better classification. An instance of related classes is 'interest' and 'money-fx' the result of classification success rate of the two classes against each other is brought in figure 4-3. This example shows that when there are similarities between distinct classes, LGDA offers a more accurate classification than LDA does. In figure 4-2 we compare the total success rate of the two models again it can be seen that in the majority of instances LGDA offers either comparable or improved results in comparison to LDA. That again coincides with our expectation that LGDA acts like LDA under certain circumstances. In figure 4-4 we bring the total classification accuracy of the two models. We need to emphasize the difference between figures 4-2 and 4-4. While figure 4-2 shows the class by class comparison of success rates, figure 4-4 shows the total success rate of the two models. In order to suppress the effects of over compensation by different classes in derivation of figure 4-4 we limited the number of documents in each class to 1000. Table 4.2 shows the total confusion matrix of the LGDA model for the optimal case.

4.1.6 Natural scene classification

The application of the LDA model to natural scene classification was first proposed by Fei. Fei et al. in [31]. In their work certain adaptation were proposed to make the

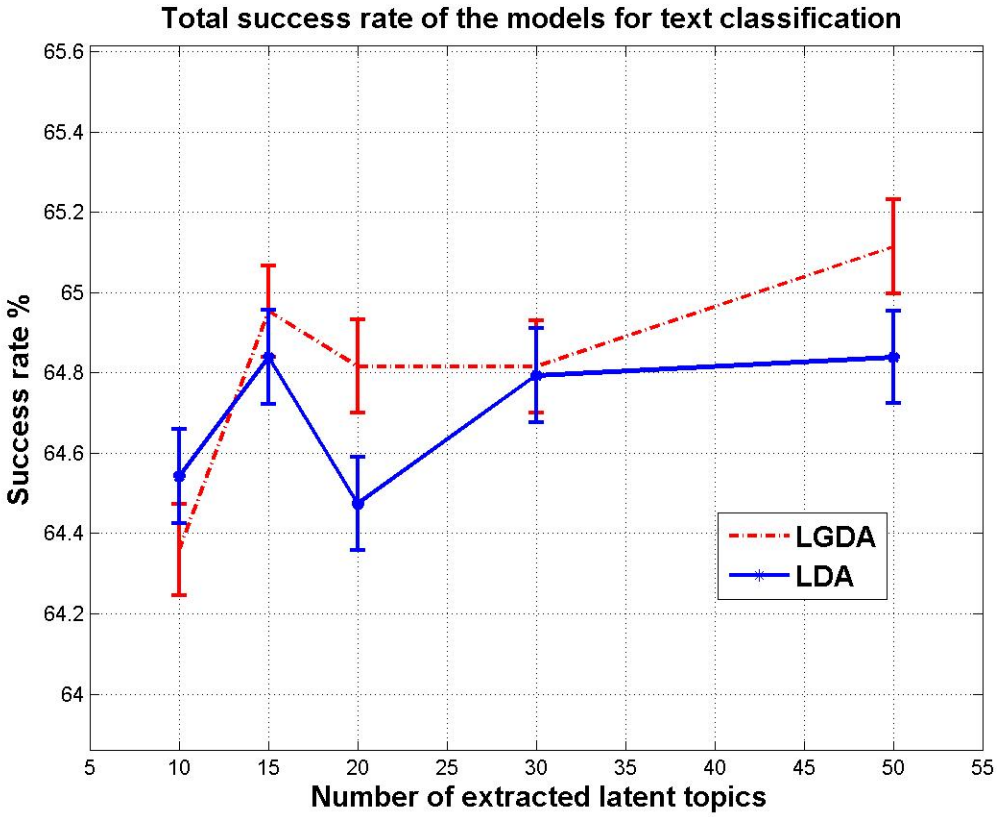


Figure 4-4: Total classification success rate. Red line: LGDA, blue Line: LDA

	acq	crude	earn	grain	interest	money-fx
acq	718	16	171	0	31	8
crude	149	507	130	12	40	61
earn	25	17	445	6	8	20
grain	33	14	81	554	28	25
interest	26	11	80	6	252	238
money-fx	49	14	93	15	120	374

Table 4.2: Confusion matrix of the LGDA model in the optimal case.

	acq	crude	earn	grain	interest	money-fx
acq	720	16	170	2	31	11
crude	150	510	132	12	36	61
earn	26	14	446	6	8	23
grain	30	12	79	554	28	29
interest	26	11	80	5	273	267
money-fx	48	16	93	14	103	335

Table 4.3: Confusion matrix of the LDA model in the optimal case.

model applicable to scene classification. We assert that our proposed model include those adaptations without a need for further assumptions. In order to derive the count vectors they used the Scale invariant feature transform (SIFT) descriptors [53] and applied it over the training set to extract the training set features. In the next step they proceeded by extracting different numbers of visual words through clustering the training feature set using K-means algorithm and assigning the centrodies of the clusters to the visual words. At the end count data are generated by assigning each of the training features to the nearest visual word generated in the last step. This approach for generating count data from extracted visual descriptors was first mentioned in [27] and is considered as a well established approach.

In our work we proceeded with applying the same method over the same natural scene database as in [31]. Samples of the dataset are shown in figure 4-5. We, however, faced a problem that was not addressed in [31]. The authors of [31] started their visual keywords from as low as 20. This poses a major problem in LDA and concordantly in LGDA as well. LDA and LGDA models are basically designed as text generation models. One would expect them to work most efficiently when the analyzed data resembles that of a text more closely. One quality of the text documents that was the center of attention in the original work [12] was the assumption that the number of count data extracted from each document in relation to the available vocabulary was considerably small and thus resulting in sparse training vectors. Our experiments did not show the same results for the scene classification work. In fact we proceeded with applying the standard implementation of the LDA algorithm and our own LGDA implementation yet both models failed to provide reliable training before the visual words number was increased to as high as 500. Computation limitations prevented us from extracting more than 1000 visual words in which extend 6 class out of the 13 class of the dataset in [31] were properly trained for both the standard LDA and the LGDA models. Regarding the work in [31] we assume that they used a preprocessing, that we are not aware of, which allowed them to overcome the non sparse nature of their training data. In this work therefore we shall proceed with showing the results of applying the two models over the portions of the dataset in [31] which both models

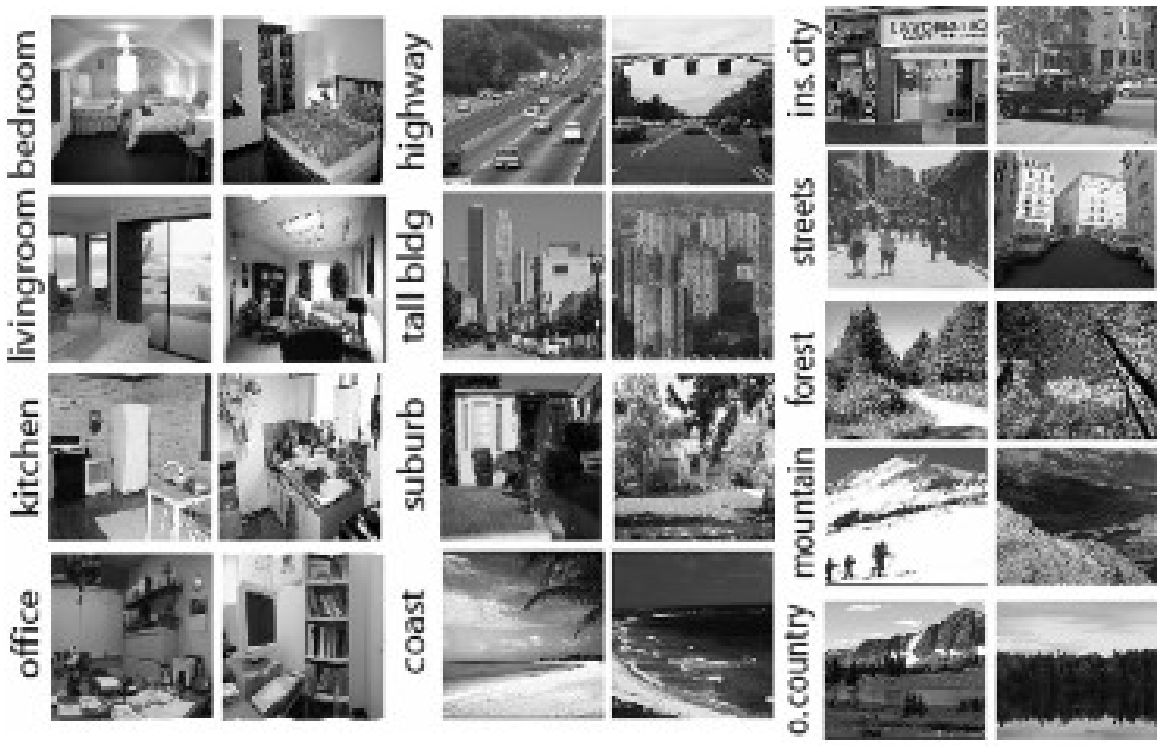


Figure 4-5: Samples of the natural scene dataset [31].

succeeded in properly learning. Figure 4-15 shows the results of applying the two models over different numbers of extracted latent topics. Figure 4-14 shows the total success rate of the model. It is understood that due to the far less sparse nature of the visual count data vectors in comparison to the text count data ones, the better fitting nature of the LGDA in comparison to LDA must be more evident. This improvement is further elaborated in figure 4-14, as it can be seen that LGDA performs clearly superior to LDA in natural scene classification. Table 4.13 shows the total confusion matrix of the LGDA model for the optimal case.

	Coast	**Forest**	**Highway**	**Inside of cities**	**Opencountry**	**Streets**	**Tall building**	**Bedroom**
Coast	223	0	51	3	50	1	2	1
Forest	1	192	10	12	0	23	0	0
Highway	33	2	104	6	18	8	14	1
Inside of cities	0	25	8	173	0	26	0	0
Open country	76	30	7	13	321	10	18	11
Streets	0	57	41	15	0	173	0	0
Tall building	9	0	3	22	7	3	268	12
Bedroom	16	18	30	59	11	42	51	185

Table 4.4: Optimal confusion matrix of the LGDA model applied for the scenes classification task.

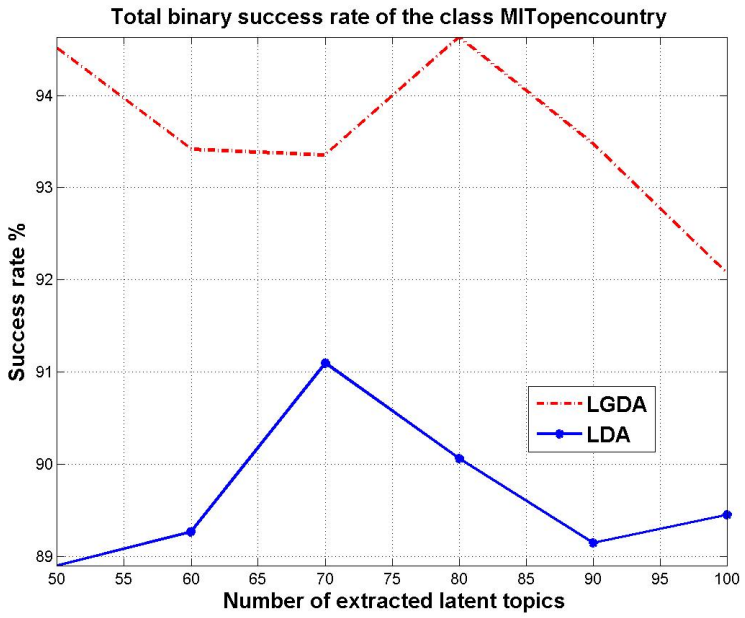


Figure 4-6: Comparison of binary classification success rate of the two models for natural scene classification. Red line: LGDA, blue Line: LDA

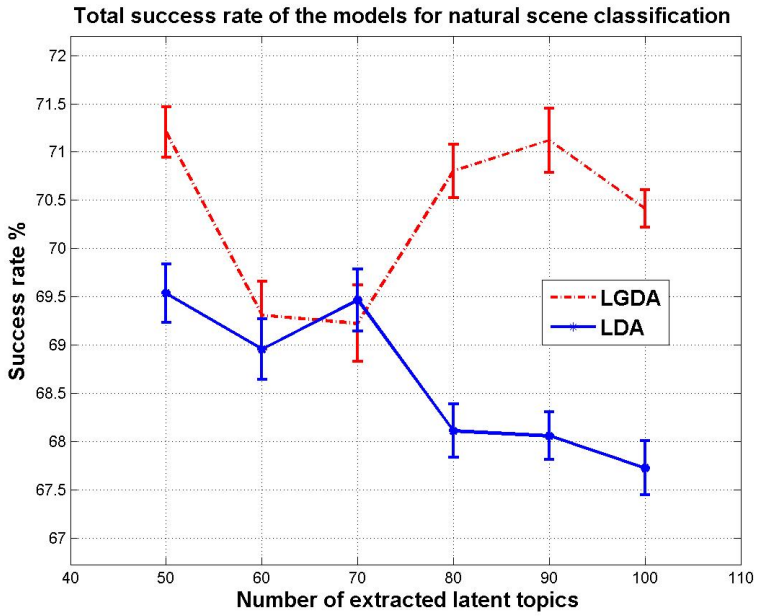
	Coast	Forest	Highway	Inside of cities	Open country	Streets	Tall building	Bedroom
Coast	231	0	59	2	53	1	1	1
Forest	1	187	5	16	0	21	0	0
Highway	55	10	100	21	47	19	62	4
Inside of cities	0	21	6	168	0	22	0	0
Open country	49	27	5	7	289	6	13	7
Streets	1	53	49	16	0	178	0	0
Tall building	5	0	2	13	4	2	219	6
Bedroom	16	28	29	61	15	40	58	192

Table 4.5: Optimal confusion matrix of the LDA model applied for the scenes classification task.

4.1.7 Comparison of the computational requirements of the LGDA versus LDA models

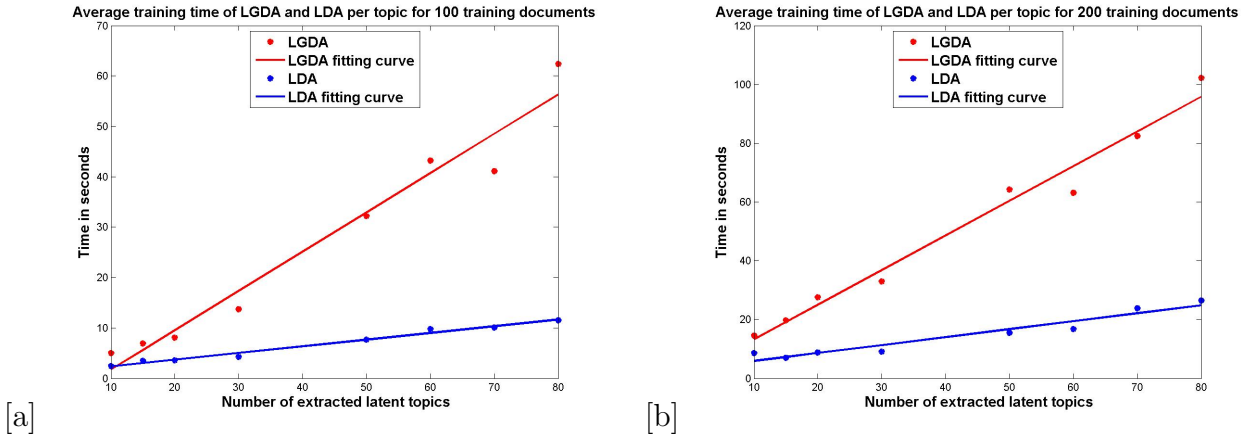
An essential concern with proposing new models as replacements for already established ones is the trade off between what the model offers and what it requires in return. LGDA in general is a more computationally demanding model than LDA. The number of parameters that need to be estimated to derive the variational and model generalized Dirichlet distributions in LGDA for the same number of latent topics is twice the numbers needed for LDA. The other parameters remain the same. One concern is the computation requirements of the model parameter estimation re-

[a]



[b]

Figure 4-7: Comparison of classification success rate of the models for natural scene classificaiton dataset.[a] 500 extracted keywords. [b] 1000 extracted keywords. Red line: LGDA, blue Line: LDA



[a] [b]

Figure 4-8: Comparison of the computation time needed for training the two models for different number of training documents. Red line: LGDA, blue Line: LDA

garding the inversing of the Hessian matrix in both models. It was shown in this chapter that like the case of LDA the computation of the Hessian matrix in LGDA is a linear process in relation to the number of generalized Dirichlet parameters.
To show the computational requirements of our model in comparison with LDA we proceed with performing a series of experiments depicting the time it takes for both models to learn their parameters in different learning conditions. The result of the experiments is shown in figure 4-8. As it can be seen from 4-8 even though in general LGDA is a more computationally demanding model, like LDA, the computation demand for additional extracted topics, clearly follow a linear curve.

4.2 A Latent Topic Model Based on the Beta-Liouville Distribution

4.2.1 Introduction

documents are mixtures of topics, where a topic is a probability distribution over words. PLSA has been applied in a variety of applications (see, for instance, [73, 36]). Several other topic models have been proposed to generalize PLSA by making slightly different statistical assumptions [80, 76]. One of the most cited extensions, improving

upon PLSA, is the latent Dirichlet allocation (LDA) model proposed in [12]. The main idea behind LDA is based on the fact that the PLSA model does not make any assumption about how the mixtures of topics weights are generated and then cannot assign likelihoods to unseen documents [21]. Thus, to improve the generalization capabilities of PLSA, the authors in [12] proposed the consideration of Dirichlet priors on the mixture weights. LDA has been successfully used to tackle problems including semantic representation [37], natural scenes categorization [31], and text classification [12].

The goal of this section is to propose an extension of LDA based on exploiting the interesting properties of the BL distribution. To maintain consistency with the original LDA model we call our model latent Beta-Liouville allocation (LBLA). We shall develop a variational Bayes approach to learn the parameters of the LBLA model. The adoption of variational Bayes is mainly motivated by the excellent results obtained when using this learning approach in several machine learning problems [45] in general and in the case of LDA in particular [12]. The Dirichlet distribution is a special case of the Beta-Liouville [15], therefore it is expectable that the LBLA will provide good modeling capabilities. Indeed, we shall elaborate the conjunctions between the two models further through extensive simulations based on challenging real-world problems.

4.2.2 Latent Beta-Liouville Allocation

The Model

Like LDA, LBLA is a fully generative probabilistic model over a corpus. A corpus in our case is a collection of M documents (or images) denoted by $\boldsymbol{M} = (\boldsymbol{w_1}, \boldsymbol{w_2}, \ldots, \boldsymbol{w_M})$. Each document $\boldsymbol{w}_m$ is represented as a sequence of N_m words $\boldsymbol{w}_m = (w_{m1}, \ldots, w_{mN_m})$. This representation is common in several applications such as text indexing [70, 84, 83]. In what follows, for sheer convenience, we drop the index m wherever we are not referring to a specific document. The word $w_n = (w_n^1, \ldots, w_n^V)$ is considered as a

binary vector drawn from a vocabulary of V words, so that $w_n^j = 1$ if the $j-th$ word is chosen and zero, otherwise. The model proceeds with generating every single word (or visual word) of the document (or the image) through the following steps:

1. Choose $N \propto Poisson(\zeta)$.

2. Choose $(\theta_1, \ldots, \theta_d) \propto BL(\vec{\xi})$.

3. For each of the N words w_n:
   (a) choose a topic $z_n \propto Multinomial(\vec{\theta})$.
   (b) Choose a word w_n from $p(w_n|z_n, \beta_w)$.

In above z_n is a $D+1$ dimensional binary vector of topics defined so that $z_n^i = 1$ if the $i-th$ topic is chosen and zero, otherwise. We define, $\vec{\theta} = (\theta_1, \ldots, \theta_{D+1})$, where $\theta_{D+1} = 1 - \sum_{i=1}^{D} \theta_i$. A chosen topic is attributed to a multinomial prior β_w over the vocabulary of words so that $\beta_{w_{(ij)}} = p(w^j = 1|z^i = 1)$, from which every word is randomly drawn. $p(w_n|z_n, \beta_w)$ is a multinomial probability conditioned on z_n and $BL(\vec{\xi})$ is a D-variate Beta-Liouville distribution with parameters $\vec{\xi} = (\alpha_1, \alpha_2, \ldots, \alpha_D, \alpha, \beta)$ and probability distribution function given by:

$$P(\theta_1, \ldots, \theta_D|\vec{\xi}) = \frac{\Gamma(\sum_{d=1}^{D} \alpha_d)\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \prod_{d=1}^{D} \frac{\theta_d^{\alpha_d - 1}}{\Gamma(\alpha_d)} \left(\sum_{d=1}^{D} \theta_d\right)^{\alpha - \sum_{l=1}^{D} \alpha_l} \times \left(1 - \sum_{l=1}^{D} \theta_l\right)^{\beta - 1} \tag{4.23}$$

It is straightforward to show that when $\beta_d = \alpha_{(d+1)} + \beta_{(d+1)}$, the Beta-Liouville distribution is reduced to Dirichlet distribution [15]. The mean, the variance, and the covariance in the case of the Beta-Liouville are as follows [15]:

$$E(\theta_d) = \frac{\alpha}{\alpha+\beta} \frac{\alpha_d}{\sum_{d=1}^{D} \alpha_d} \tag{4.24}$$

$$var(\theta_d) = \left(\frac{\alpha}{\alpha+\beta}\right)^2 \frac{\alpha_d(\alpha_d+1)}{(\sum_{m=1}^{D} \alpha_d)(\sum_{m=1}^{D} \alpha_m + 1)} - E(\theta_d)^2 \frac{\alpha_d^2}{(\sum_{m=1}^{D} \alpha_m)^2} \tag{4.25}$$

and the covariance between θ_i and θ_j is given by:

$$Cov(\theta_l, \theta_k) = \frac{\alpha_l \alpha_k}{\sum_{d=1}^{D} \alpha_d} \left(\frac{\frac{\alpha+1}{\alpha+\beta+1}\frac{\alpha}{\alpha+\beta}}{\sum_{d=1}^{D} \alpha_d + 1} - \frac{\frac{\alpha}{\alpha+\beta}}{\sum_{d=1}^{D} \alpha_d} \right) \tag{4.26}$$

It can be seen from the previous equation, that the covariance matrix of the Beta-Liouville distribution is more general than the covariance matrix of the Dirichlet distribution which is strictly negative. Moreover, unlike Dirichlet, two elements with the same mean value can have different variances. Beta-Liouville distribution, like the Dirichlet distribution, belongs to the exponential family of distributions (see Appendix 1). This means that the Beta-Liouville distribution has a conjugate prior that can be developed in a formal way, which is an important property that we shall use in the following for the learning of our model. It turns out also that Beta-Liouville, like Dirichlet, is a conjugate prior of the multinomial distribution. This implies that if $(\theta_1, \ldots, \theta_D)$ follows a Beta-Liouville distribution with parameters $\vec{\xi}$, and $\vec{N} = (n_1, \ldots, n_{D+1})$ follows a multinomial with parameter $\vec{\theta}$, then the posterior distribution $p(\vec{\theta}|\vec{\xi}, \vec{N})$ also follows a Beta-Liouville distribution with parameters $\xi^{'}$ given as follows [15]:

$$\alpha_d^{'} = \alpha_d + n_d \quad \alpha' = \alpha + \sum_{d=1}^{D} n_d \quad \beta' = \beta + n_{D+1} \tag{4.27}$$

Having our Beta-Liouville prior in hand, we proceed with defining the $(D+1) \times V$ word-topic probability matrix β_w which element $\beta_{w_{ij}} = p(w_j = 1|z_i = 1)$ shows the probability of drawing the $j-th$ word given that the $i-th$ latent topic is chosen. Like the LDA case, we proceed with assuming a non-generated β_w matrix, but we will show that this assumption does not have a serious impact and it can be revoked without bringing harm to the entire model. By assuming conditional independence of the variables, the same as LDA, one can deduce the following joint distribution:

$$p(\vec{\theta}, \boldsymbol{z}, \boldsymbol{w}|\vec{\xi}, \beta_w) = p(\vec{\theta}|\vec{\xi}) p(\underline{o}ldsymbolw|\boldsymbol{z}, \beta_w) p(\boldsymbol{z}|\vec{\theta}) \tag{4.28}$$

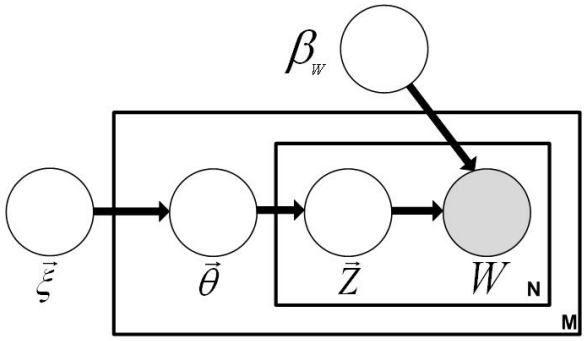


Figure 4-9: Graphical representation of LBLA model. The shaded circles show observed nodes. The blank circles are the hidden nodes. From outside to inside is the corpus space, the document space and the word space.

where $\boldsymbol{z}$ is the set of latent topics. Integrating over the $\vec{\theta}$ parameters and the topic space gives

$$\begin{aligned} P(\boldsymbol{w}|\vec{\xi},\beta_w) &= \int \prod_{n=1}^{N} \frac{\Gamma(\sum_{d=1}^{D}\alpha_d)\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \prod_{d=1}^{D} \frac{\theta_d^{\alpha_d-1}}{\Gamma(\alpha_d)} \left(\sum_{l=1}^{D}\theta_l\right)^{\alpha-\sum_{l=1}^{D}\alpha_l} \\ &\times \left(1-\sum_{l=1}^{D}\theta_l\right)^{\beta-1} \prod_{i=1}^{D+1}\prod_{j=1}^{V}(\theta_i\beta_{W_{ij}})^{w_n^j} d\theta \end{aligned} \tag{4.29}$$

In the previous equation, $\vec{\xi}$ and β_w are the corpus level parameters that are selected once per each document in the corpus. $\vec{\theta}$ is the document level parameter and is chosen once per document. $\boldsymbol{z}$ and $\boldsymbol{w}$ are word level parameters and are chosen once per every word inside each document. Thus, we can obtain the probability of the corpus as follows:

$$p(\boldsymbol{D}|\vec{\xi},\beta_w) = \prod_{m=1}^{M} p(\boldsymbol{w}_m|\vec{\xi},\beta_w) \tag{4.30}$$

LBLA has basically the same probabilistic graphical model as LDA as it is shown in figure 4-9.

LBLA Inference

The main inference problem of LBLA is estimating the posterior of the hidden variables, $\vec{\theta}$ and $\boldsymbol{z}$:

$$p(\vec{\theta}, \boldsymbol{z}|\boldsymbol{w}, \vec{\xi}, \beta_w) = \frac{p(\vec{\theta}, \boldsymbol{z}, \boldsymbol{w}|\vec{\xi}, \beta_w)}{p(\boldsymbol{w}|\vec{\xi}, \beta_w)} \tag{4.31}$$

The above equation is known to be intractable. An efficient way to estimate the parameters in this intractable posterior is to use variational Bayes (VB) inference [12]. VB inference offers a solution to the intractability problem by determining a lower bound on the log likelihood of the observed data which is mainly based on considering a set of variational distributions on the hidden variables [46, 85]:

$$q(\vec{\theta}, \boldsymbol{z}|\boldsymbol{w}, \vec{\xi_q}, \Phi_{\boldsymbol{w}}) = q(\vec{\theta}|\vec{\xi_q}) \prod_{n=1}^{N} q(z_n|\phi_n) \tag{4.32}$$

In the above $q(\vec{\theta}|\vec{\xi_q})$ can be viewed as a variational Beta-Liouville distribution, calculated once per document, $q(z_n|\phi_n)$ is a multinomial distribution with parameter ϕ_n extracted once for every single word inside the document, and $\Phi_{\boldsymbol{w}} = \{\phi_1, \phi_2, \ldots, \phi_N\}$. Using Jensen's inequality [46] one can derive the following:

$$\log p(\boldsymbol{w}|\vec{\xi}, \beta_w) \geq E_q[\log p(\vec{\theta}, \boldsymbol{z}, \boldsymbol{w}|\vec{\xi}, \beta_w)] - E_q[\log q(\vec{\theta}, \boldsymbol{z})] \tag{4.33}$$

Assigning $L(\vec{\xi_q}, \Phi_{\boldsymbol{w}}; \vec{\xi}, \beta_w)$ to the right-hand side of the above equation it can be shown that the difference between the left-hand side and the right-hand side of the equation is the KL divergence between the variational posterior probability and the actual posterior probability, thus we have:

$$\log p(\boldsymbol{w}|\vec{\xi}, \beta_w) = L(\vec{\xi_q}, \Phi_{\boldsymbol{w}}; \vec{\xi}, \beta_w) + KL\big(q(\vec{\theta}, \boldsymbol{z}|\vec{\xi_q}, \Phi_{\boldsymbol{w}})||p(\vec{\theta}, \boldsymbol{z})|\boldsymbol{w}, \vec{\xi}, \beta_w)\big) \tag{4.34}$$

The left hand side of the above equation is constant in relation to variational parameters, therefore to minimize the KL divergence on the right-hand side one can proceed with maximizing $L(\vec{\xi_q}, \Phi_{\boldsymbol{w}}; \vec{\xi}, \beta_w)$. Up to here the formulation basically follows the LDA model. The divergence of the models begins when we proceed with assigning the

Beta-Liouville distribution as the parameter generator instead of the LDA Dirichlet assumption. In appendix 2 we bring the breakdown of $L(\vec{\xi_q}, \Phi_{\boldsymbol{w}}; \vec{\xi}, \beta_w)$.
Using variational inference to maximize the lower bound $L(\vec{\xi_q}, \Phi_{\boldsymbol{w}}; \vec{\xi}, \beta_w)$ with respect to ϕ_{nl}, we derive the following updating equations for the variational multinomial (see Appendix 2.1)

$$\phi_{nl} = \beta_{lv} e^{(\lambda_n - 1)} e^{(\Psi(\gamma_i) - \Psi(\sum_{ii=1}^{D} \gamma_{ii}))} \tag{4.35}$$

$$\phi_{n(D+1)} = \beta_{(D+1)v} e^{(\lambda_n - 1)} e^{(\Psi(\beta_\gamma) - \Psi(\alpha_\gamma + \beta_\gamma))} \tag{4.36}$$

where Ψ is the digamma function, $\beta_{lv} = p(w^v = 1 | z^l = 1)$ and the weighing constant $e^{\lambda_n - 1}$ is given by:

$$e^{\lambda_n - 1} = \frac{1}{\beta_{(D+1)v} e^{(\Psi(\beta_\gamma) - \Psi(\alpha_\gamma + \beta_\gamma))} + \sum_{i=1}^{D} \beta_{iv} e^{(\Psi(\gamma_d) - \Psi(\sum_{l=1}^{D} \gamma_l))}} \tag{4.37}$$

Maximizing the lower bound L with respect to the variational Beta-Liouville parameters gives the following updating equations (see Appendix 2.2):

$$\gamma_i = \alpha + \sum_{n=1}^{N} \phi_{ni} \quad \alpha_\gamma = \alpha + \sum_{n=1}^{N} \sum_{d=1}^{D} \phi_{nd} \quad \beta_\gamma = \beta + \sum_{n=1}^{N} \phi_{n(D+1)} \tag{4.38}$$

Comparing the above equations with equation 4.27 shows that the variational Beta-Liouville for each document acts as a posterior in the presence of the variational multinomial parameters. The same conclusion was observed in [12] for the LDA case. This is a direct result of the conjugacy between the Beta-Liouville and the multinomial.

Parameters Estimation

The goal of this subsection is to find the model's parameters estimates based on the variational parameters derived in the last subsection. One needs to consider that the LBLA parameters are corpus parameters and therefore they are estimated by considering all M documents inside the corpus. In the following, we denote $L =$

$\sum_{m=1}^{M} L_m$ as the lower bound corresponding to all the corpus, where L_m is the lower bound corresponding to each document m.

Maximizing the corpus lower bound L with respect to $\beta_{w(lj)}$ delivers the following updating equation (see Appendix A.6)

$$\beta_{w(lj)} \propto \sum_{d=1}^{M} \sum_{n=1}^{N_d} \phi_{dnl} w_{dn}^{j} \tag{4.39}$$

The model's parameters are the last ones to be derived. Following the work of Minka [56], it was shown in [12] that in order to derive LDA parameters it was feasible to use the Newton-Raphson algorithm for parameters estimation. It was also shown that due to the characteristics of the Dirichlet distribution, it is possible to exchange the computationally demanding problem of inverting the Hessian matrix of the Lower bound with a linear operator and therefore reducing the model complexity.

The Hessian matrix of the Beta-Liouville distribution is derived quite similarly to the one of the Dirichlet distribution. The main difference is the addition of an extra 2×2 matrix inversion for the derivation of the Beta parameters (α and β). The complete derivation of the model parameters is brought in apendix 4. The last formulation that we need to derive to prepare our model for the classification task is the likelihood of a document in our model. This can be done by deriving first the likelihood of a randomly chosen word w_n inside the document:

$$\begin{aligned} p(w_n|\vec{\xi}) &= \sum_{l=1}^{D+1} \int p(w_n|z_l) p(z_l|\vec{\theta}) p(\vec{\theta}|\vec{\xi}) d\vec{\theta} \\ &= \sum_{l=1}^{D+1} p(w_n|z_l) \int p(z_l|\vec{\theta}) p(\vec{\theta}|\vec{\xi}) d\vec{\theta} \\ &= \sum_{l=1}^{D} \beta_{l(v|w_n^v=1)} E[\theta_l] + \beta_{(D+1)(v|w_n^v=1)} (1 - \sum_{l=1}^{D} E[\theta_l]) \end{aligned} \tag{4.40}$$

By substituting the value of $E[\theta_l]$ (see Eq. 4.24) into Eq. 4.40 we obtain the formulation for the word likelihood as follows:

$$p(w_n|\vec{\xi}) = \sum_{l=1}^{D} \beta_{l(v|w_n^v=1)}(\frac{\alpha}{\alpha+\beta}\frac{\alpha_D}{\sum_{d=1}^{D}\alpha_d}) + \beta_{(D+1)(v|w_n^v=1)}(1 - \sum_{l=1}^{D}(\frac{\alpha}{\alpha+\beta}\frac{\alpha_d}{\sum_{d=1}^{D}\alpha_d})) \tag{4.41}$$

The log likelihood of a document $\boldsymbol{w}$ is derived as the sum of the log likelihoods of the words present inside the document and therefore we have:

$$\log p(\boldsymbol{w}_m|\vec{\xi}) = \sum_{n=1}^{N_{\boldsymbol{m}}} \log p(w_n|\vec{\xi}) = \sum_{v=1}^{V} cnt_{mv} \log p(w_v|\vec{\xi}) \tag{4.42}$$

where for each document $\boldsymbol{w}_m$, cnt_{mv} is the number of times the $v-th$ word is drawn.

4.2.3 Experimental Results

In this section, we investigate the LBLA model on three distinct challenging applications namely text and visual scene classification, and action recognition. The main goal of these three applications is to compare the LBLA and LDA performances.

Text Classification

most approaches can be looked upon from two distinct but related ways. Assuming that the number of classes is known, from a first perspective text classification can be viewed as a binary categorization problem where the main task is to decide to which class we should assign the text given two distinctively chosen classes. The other way to look upon the problem is to decide how accurately the model can assign the proper class to a document at the presence of all other classes. We will consider the two mentioned scenarios in the following.

For our simulations, we again choose the Reuters-21578 dataset [2] [44]. We limit ourselves to the top 6 categories extracted from the dataset. They, in total, comprise

[2] http://www.daviddlewis.com/resources/testcollections/reuters21578/

more than 9000 documents of the original dataset and nearly all the words present in the unabridged dataset. Table 4.1 describes the considered classes.

To examine the classification accuracy of the models, in the first step we choose a certain number of the documents in each of the classes as training documents. Next, we learn our models for each of the chosen training sets, for different numbers of latent topics to observe the effect of choosing them on the classification accuracy. Classification in this first experiment is regarded as a binary process meaning that a given document is presented to two different trained classes and the class that gives the higher likelihood is chosen as the document class. Selected success rates of the two models under several training conditions are brought in figure 4-10. An interesting observation regarding the two models can be deduced from this figure. The Reuters dataset consists of relatively short documents that are presented as extremely sparse count vectors over the entire vocabulary set. Both LDA and LBLA use variational Bayes inference as their core learning method; however the sparsity of the count vectors causes both models to basically provide the same fit over the training set. The result is that facing sparse vectors, the two models roughly offer the same success rate. This can be seen in figure 4-10. There is an exception to

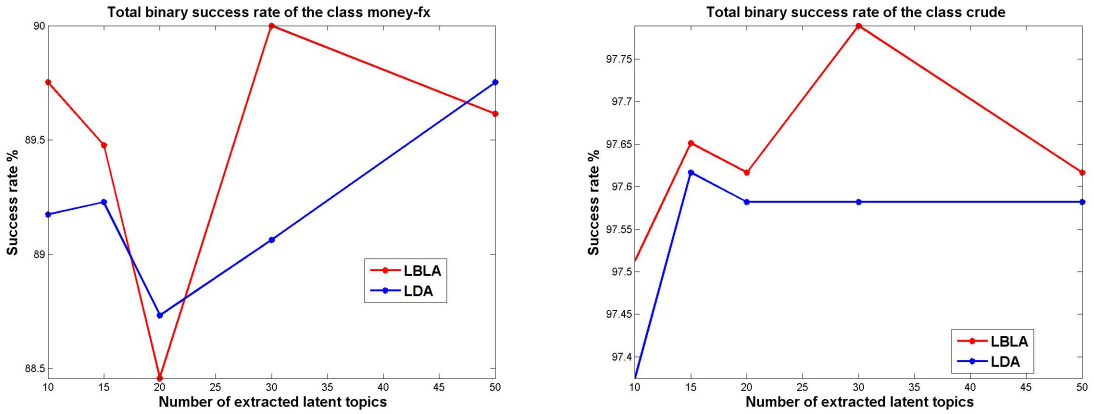


Figure 4-10: Examples of binary classification success rates of the LBLA and LDA models when applied for text classification. Red line: LBLA, blue line: LDA.

this observation. When the two classes are similar to each other, one may expect that the models fail to separate them as precisely as when the classes are dissimilar.

In this case the model that offers the better fitting to its training set could offer better classification. An instance of related classes is 'interest' and 'money-fx'; the classification success rates obtained when using LBLA and LDA, in this case, are displayed in figure 4-11.

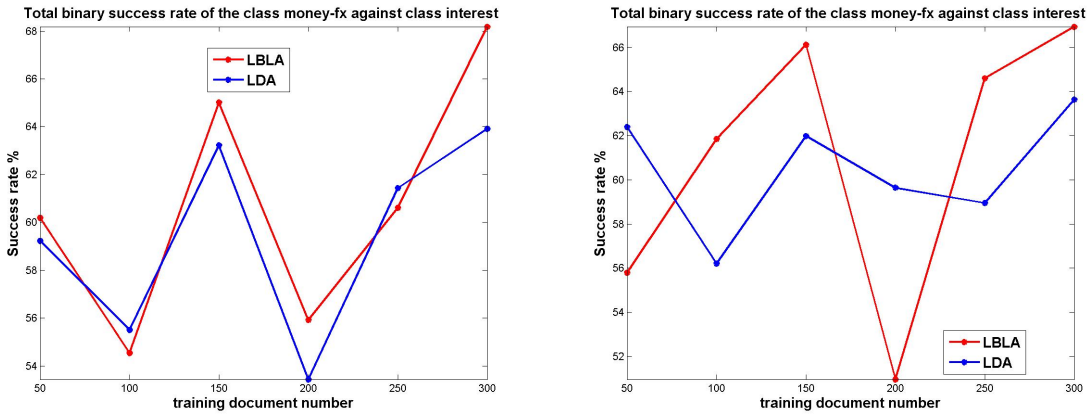


Figure 4-11: Comparison of binary classification success rates of the LBLA and LDA models for 'Money-fx' class against 'interest' class when we consider (a) 15 extracted latent topics, and (b) 30 extracted latent topics. Red line: LBLA, blue line: LDA.

This example shows that when there are similarities between distinct classes, LBLA offers a more accurate classification than LDA. Thus, we can conclude, according to figures 4-10 and 4-11, that in the majority of cases LBLA offers either comparable or improved results as compared to LDA. That again coincides with our expectation that LBLA acts like or better than LDA.

In figure 4-12, we compare the total classification accuracies of the two models. We need to emphasize the difference between figures 4-10 and 4-12. While figure 4-10 shows the class by class comparison of success rates, figure 4-12 shows the total success rate of the two models. In order to suppress the effects of over compensation by different classes in derivation of figure 4-12 we limited the number of documents in each class to 1000. Tables 4.6 and 4.7 show the confusion matrices of the LBLA and LDA models, respectively, for the optimal cases (i.e. corresponding to the maximum rates in figure 4-12).

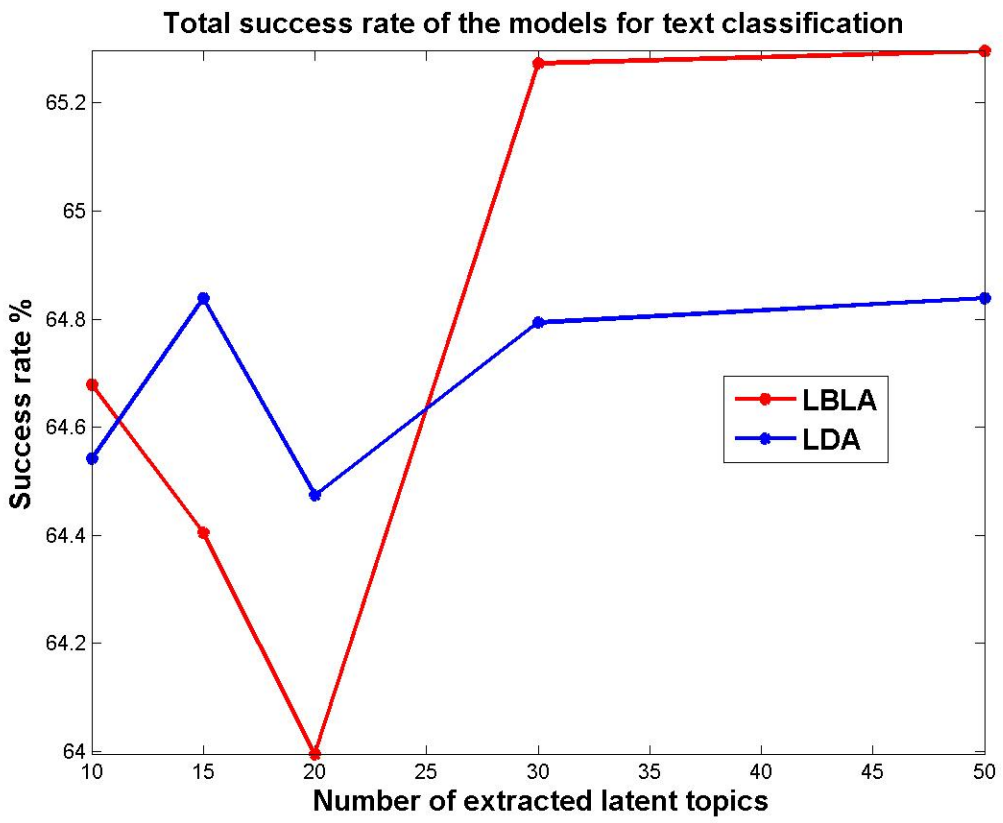


Figure 4-12: Total text classification success rates obtained using LBLA and LDA models. Red line: LBLA, blue line: LDA.

Table 4.6: Confusion matrix of the LBLA model, in the optimal case, when applied to text classification.

	acq	crude	earn	grain	interest	money-fx
acq	718	15	167	3	31	9
crude	149	515	130	15	39	61
earn	26	15	452	6	7	22
grain	32	13	79	546	27	25
interest	26	8	81	7	263	246
money-fx	49	13	91	16	112	363

Visual Scenes Classification

Methodology

In this set of experiments, we apply our LBLA model to the challenging task of visual scenes classification which has attracted a lot of attention recently [27, 31, 52, 50, 20].

Table 4.7: Confusion matrix of the LDA model, in the optimal case, when applied to text classification.

	acq	crude	earn	grain	interest	money-fx
acq	720	16	170	2	31	11
crude	150	510	132	12	36	61
earn	26	14	446	6	8	23
grain	30	12	79	554	28	29
interest	26	11	80	5	273	267
money-fx	48	16	93	14	103	335

The main goal is to compare the LBLA to the LDA which was considered for the same task in [31]. It is noteworthy that some adaptations to the original LDA were proposed in [31], and the reader is then referred to this paper for more details, to make it applicable to scenes classification. The very same adaptations were included in the LBLA without a need for further assumptions. The main idea that we use here is based on the description of scenes using visual words [27]. This approach has emerged over the past few years and received strong interest that is mainly motivated by the fact that many of the techniques previously proposed for text classification can be adopted for images categorization [27, 87, 77].

For the construction of the visual words vocabulary, we need first to extract local descriptors from a set of training images. Many descriptors have been proposed in the past, but scale invariant feature transform (SIFT) descriptor [53], that we consider here, has dominated the literature. The extracted features are then quantized through clustering (the K-Means algorithm in our case) and the obtained d clusters centroids are considered as our visual words. Having the visual vocabulary in hand, each image can be represented as a d-dimensional vector containing the frequency of each visual word in that image. In our experiment we take 7 classes from the natural scenes dataset introduced in [71] and we combine it with one indoor scenes class from [31]. The 7 classes chosen from the data set described in [71] are coast, forest, highway, inside of cities, open country, street, and tall building, which contain 361, 329, 261, 309, 411, 293, and 356 images, respectively. The class chosen from the data set proposed in [31] is the bedroom category which contains 217 images. Examples of

images from the different considered classes are shown in figure 4-13.

Figure 4-13: Sample images from each group. (a) Highway, (b) Inside of cities, (c) Tall building, (d) Streets, (e) Forest, (f) Coast, (g) Open country, (h) Bedroom.

Results

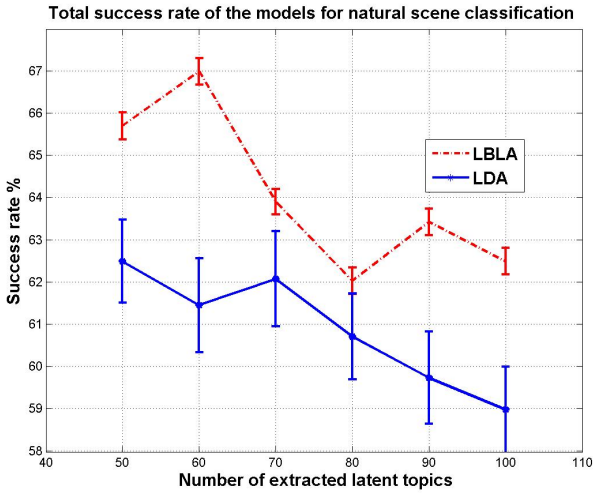


Figure 4-14: Classification success rates, as a function of the number of extracted latent topics, of the LBLA and LDA models applied for the visual scenes classification task. Red line: LBLA, blue Line: LDA.

From each category, in the considered data set, we randomly chose 100 images for model training. Unlike text classification which usually leads to sparse training matrices, the abundance of visual descriptors in the images and the relatively lower number of extracted visual keywords, as compared to the textual vocabulary, lead to less sparse matrices for scenes classification. In figure 4-14, we compare the success

rates of the LBLA and LDA models, when varying the number of extracted latent topics, over the data set. According to this figure it is clear that better categorization results are obtained when adopting LBLA. Figure 4-15 shows examples of per class comparisons between the success rates obtained by both models. It is obvious that due to the less sparse nature of the count data vectors extracted in the case of scenes classification task (as compared to the text classification task presented in the previous section), the better fitting capabilities of the LBLA become more evident. Tables 4.12 and 4.9 show the optimal confusion matrices of the LBLA and LDA mod-

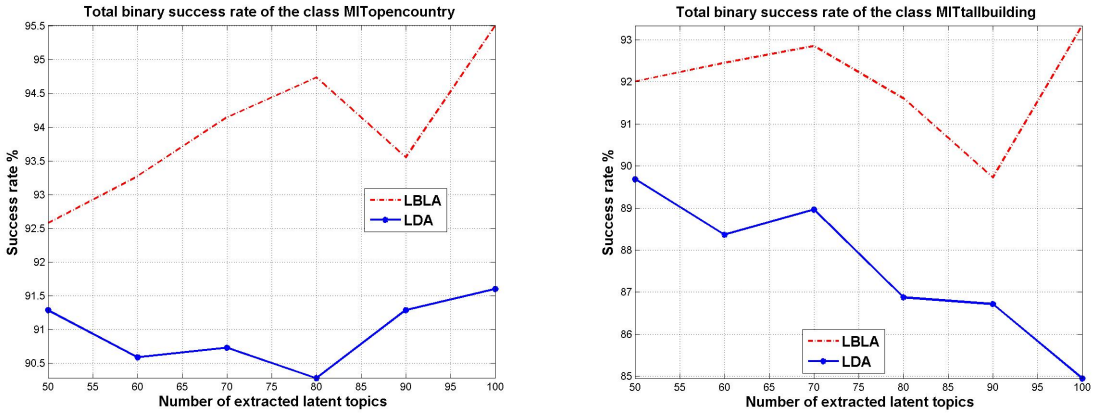


Figure 4-15: Examples of per class classification success rates, as a function of the number of extracted latent topics, of the LBLA and LDA models. Red line: LBLA, blue Line: LDA.

els. According to these tables it is clear again that the LBLA gives significantly a better classification accuracy (67.0%) than the LDA (62.49%).

	Coast	**Forest**	**Highway**	**Inside of cities**	**Open country**	**Streets**	**Tall building**	**Bedroom**
Coast	225	1	20	1	68	3	2	0
Forest	1	191	0	14	0	19	0	0
Highway	48	1	186	9	12	4	27	2
Inside of cities	0	14	0	163	0	27	0	0
Opencountry	56	52	9	13	300	10	23	7
Streets	1	44	0	16	0	174	0	0
Tall building	4	0	8	23	7	1	245	11
Bedroom	21	24	31	64	20	51	56	190

Table 4.8: Optimal confusion matrix of the LBLA model applied for the scenes classification task.

	Coast	Forest	Highway	Inside of cities	Open country	Streets	Tall building	Bedroom
Coast	231	0	59	2	53	1	1	1
Forest	1	187	5	16	0	21	0	0
Highway	55	10	100	21	47	19	62	4
Inside of cities	0	21	6	168	0	22	0	0
Open country	49	27	5	7	289	6	13	7
Streets	1	53	49	16	0	178	0	0
Tall building	5	0	2	13	4	2	219	6
Bedroom	16	28	29	61	15	40	58	192

Table 4.9: Optimal confusion matrix of the LDA model applied for the scenes classification task.

Figure 4-16: Samples of the actions used in our experiments [35].

Action Recognition

Action recognition has attracted a great deal of attention [14, 81, 35], in part because of its potential applications. For instance, it could be used in gesture recognition, video surveillance, and video indexing [25, 13, 40]. In this set of experiments, we apply the LBLA model to the action recognition problem using the space time actions dataset of [35] (see figure 4-16). The methodology of the experiments is as follows. In the first step, we applied the Horn-Schunck algorithm [43] to extract the optical flow matrix of the subsequent frames. Next, we applied an arbitrary threshold on the optical flow matrix to extract the strong optical flow responses, we used a mask of predefined size around the positions with the strong optical flow responses to form our total vector set. We used the K-means algorithm on a random 10 percent of the total vector set to extract the action flow words and having extracted them we proceeded with generating the count data vectors and training the models.

The results of applying both the LBLA and LDA models on the action recognition dataset is shown in figure 4-17. The confusion matrix for the relevant optimal case

for the LBLA and LDA models are brought in tables 4.11 and 4.10, respectively. As it can be seen from the experimental results, LBLA shows slight improvement (59.38%) in comparison to LDA (59.01%). The reason behind the slightness of the improvement can again be attributed to the sparse nature of the extracted features.

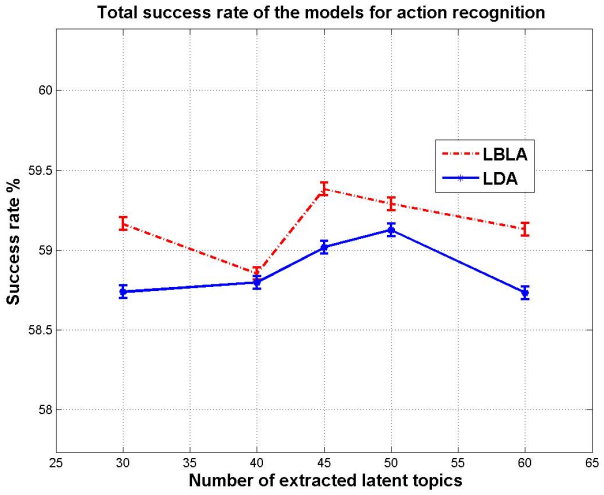


Figure 4-17: Total action recognition success rates obtained using LBDA and LDA models. Red line: LBDA, blue line: LDA.

	Jump	**Pjump**	**Run**	**Side**	**Skip**	**Walk**
Jump	512	8	35	36	29	20
Pjump	22	196	7	4	2	1
Run	47	3	352	67	55	48
Side	56	0	47	271	71	36
Skip	98	2	81	56	410	141
Walk	129	5	69	163	162	419

Table 4.10: Optimal confusion matrix of the LDA model applied for the action recognition task.

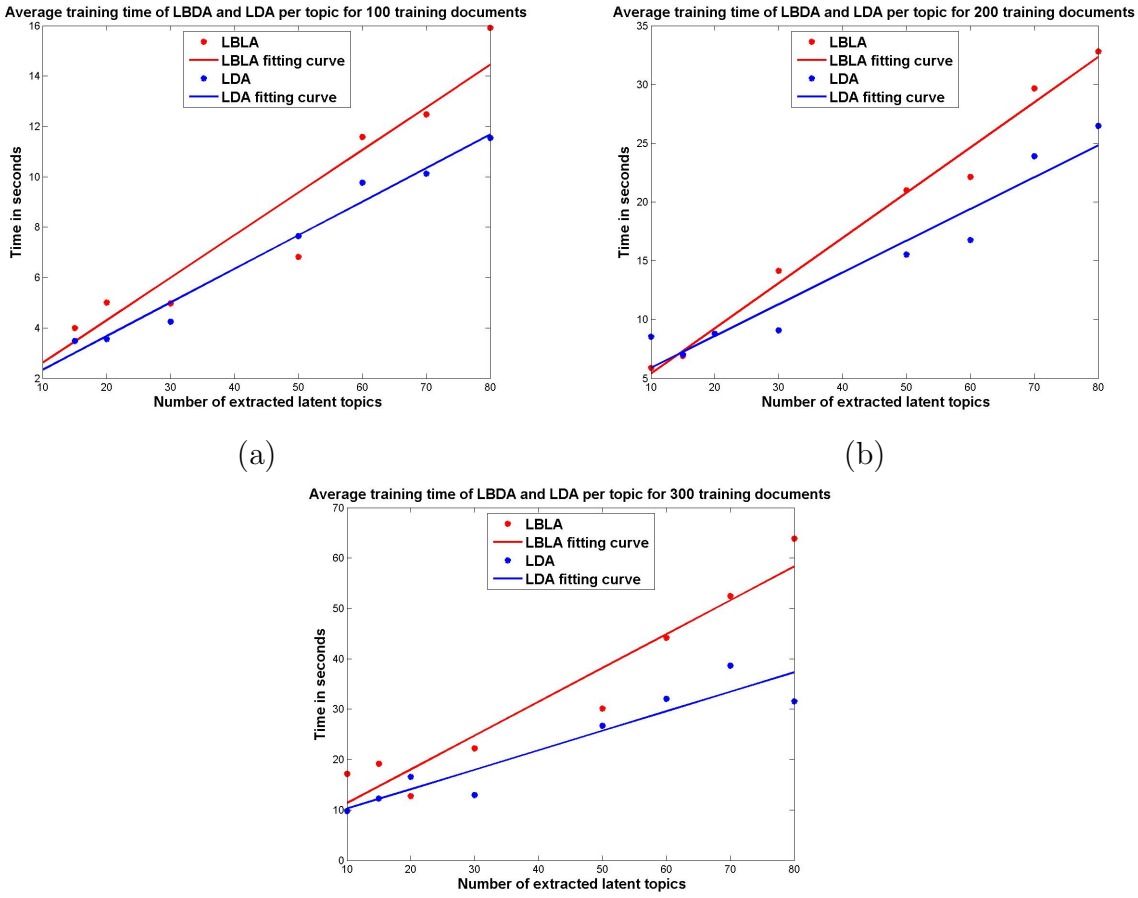
Comparison of the computational requirements of the LBLA versus the LDA

LBLA in general is a more computationally demanding model than LDA. Indeed, in dimension D the Dirichlet has $D+1$ parameters while the Beta-Liouville has $D+2$

	Jump	**Pjump**	**Run**	**Side**	**Skip**	**Walk**
Jump	510	8	31	35	29	21
Pjump	25	196	7	4	2	1
Run	49	3	362	69	54	48
Side	51	0	43	266	69	33
Skip	102	2	83	61	422	144
Walk	127	5	66	162	153	418

Table 4.11: Optimal confusion matrix of the LBLA model applied for the action recognition task.

parameters. Thus, comparing to the Dirichlet, the Beta-Liouville has only one extra parameter. The number of the other model parameters remains the same in the two models. One concern is the computational requirements of the model parameters estimation regarding the inversion of the Hessian matrix in both models. It was shown in this paper that like the LDA case the computation of the Hessian matrix in LBDA is a linearly related to the number of Beta-Liouville parameters. To show the computational requirements of our model in comparison with LDA we proceed with performing a series of experiments depicting the time it takes for both models to learn their parameters in different learning conditions. The result of these experiments are shown in figure 4-18. From this figure, we can see that although in general LBLA is a more computationally demanding model, like LDA, the computational demand for additional extracted topics, clearly follows a linear curve. the derivative of the above equations in respect to their BL parameter gives:



(a) (b) (c)

Figure 4-18: Comparison of the computational time needed for training the LDA and LBLA models, for different numbers of training documents, as a function of the number of latent topics. The numbers of considered training documents are: (a) 100, (b) 200, and (c) 300. Red line: LBLA, blue Line: LDA.

4.3 Online Learning For Topic Models

The original LDA model assumes the entire training data corpus to be available at hand [12]. This assumption poses two serious drawbacks. Firstly it mandates the need for a huge test dataset to be collected beforehand and secondly it requires huge computation resources for performing the parameter estimation over the test data. In a later work [41] Hoffman and Blei proposed an online learning model for overcoming the mentioned constraints. The model proposed in that work was called online latent Dirichlet allocation. In this chapter we shall apply the same model over our own latent topic models LGDA and LBLA and compare the two models against the online LDA model.

4.4 Online LDA model

The variational Bayes model of the LDA model is shown to converges to a local likelihood of the actual posterior of the hidden parameters of the models. However the main problem with the original VB model is that it needs to consider the entire corpus beforehand for parameter estimation. This in return emerges two serious problems. Firstly the need for the collection of the entire training corpus and secondly the computational requirements of dealing with a huge corpus. To overcome this problem Hoffman and Blei [41] offered an online learning model that fixes the mentioned issues. The solution is such that a time dependent (time defined as the index of the part of the data given to the model in each iteration) weight is defined as:

$$\rho_t = (\tau_0 + t)^{-\kappa}, \kappa \in (0.5, 1] \tag{4.43}$$

The parameter τ_0 slows down the effect of early parameter estimations. The online learning algorithm can easily be extended to cover LGDA and LBLA models as well. The steps of the algorithm are as follows.

1. In each learning interval the model performs a batch VB over the patch of the

training set attributed to that interval and assigns a weight value to the patch according to 4.43.

2. Prior parameter estimation: Perform the Newton-Raphson algorithm over the entire corpus of patches for $t = 0$ to ∞ as: $\xi \leftarrow \xi - \rho_t \widetilde{\alpha}(\xi_t)$ where $\widetilde{\alpha}(\xi_t)$ is the inverse of the Hessian times the gradient with respect to α of the posterior lower band.

3. Word dictionary update: $\widetilde{\beta_w}(t+1) = normalize((1-\rho_t)\widetilde{\beta_{wt}} + \rho_t \beta_w(t))$ where $\widetilde{\beta_w}(t)$ is the available estimation of the Word dictionary at $t-th$ step.

It was shown in [41] that the condition $\kappa \in (0.5, 1]$ is necessary for keeping the online learning model stable. The only deviation from the original online LDA model that we have done is to consider a non generative word dictionary rather than the full generative one proposed in [41]. The deviation is required so as to maintain consistency with the previous works and it was shown in this chapter that the two assumptions basically lead to similar models. In the next section we proceed with offering the experimental results of applying the LGDA and LBLA models against LDA and comparing the performance of the 3 models against each other.

4.5 Experimental Results

In this section we shall proceed with applying out proposed two models, online LBLA and LGDA, on the challenging task of natural scene classification and make a comparison between the classification success rate offered by the two models versus that of the online LDA. The main idea that we use here is based on the description of scenes using visual words. This approach has emerged over the past few years and received strong interest that is mainly motivated by the fact that many of the techniques previously proposed for text classification can be adopted for images categorization [87, 77].

We again use the bag of visual words approach for the natural scene classification task.

4.5.1 Comparison between the performance of LBLA and LGDA models against LDA

At first the models were given 5 chunks of training images each containing 20 images. In this set of experiments the effect of the online learning was reduced since the small number of iterations plus the big chunks of test data quite resembled the Batch LDA and LGDA models. The results of applying the online LDA and LBLA models are brought in figure 4-19. Under the same experimental conditions we proceed with delivering the results for the LGDA model in figure 4-20.

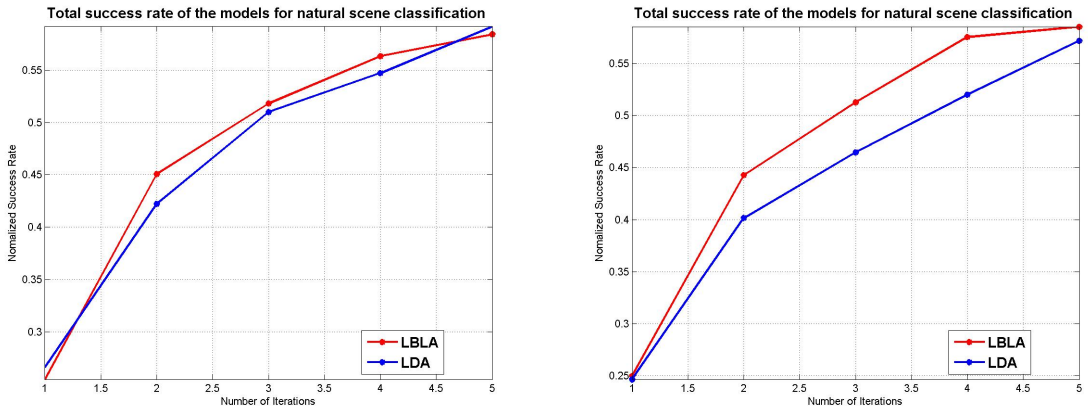


Figure 4-19: Comparison of the success rate of the online LBLA model against online LDA model for the natural scene classification, for 20 training image per step, for two different extracted number of topics.

The experiments results over the dataset show a slight advantage for the LBLA model whilst LGDA appears to stand in par with LDA. However the results become more distinguishable when we proceed with dividing the training set into yet smaller chunks. In the second set of experiments we divide the test dataset into chunk of 5 images per iteration. The result of applying the data over the LBLA versus LDA is brought in figure 4-21 The same set of experiments were performed on the LGDA model as well and the results are brought in figure 4-22 The optimal confusion matrix of the online LBLA model is brought in table 4.12. The optimal confusion matrix of the online LBLA model is brought in 4.13 and The optimal confusion matrix of the online LDA model is brought in 4.9.

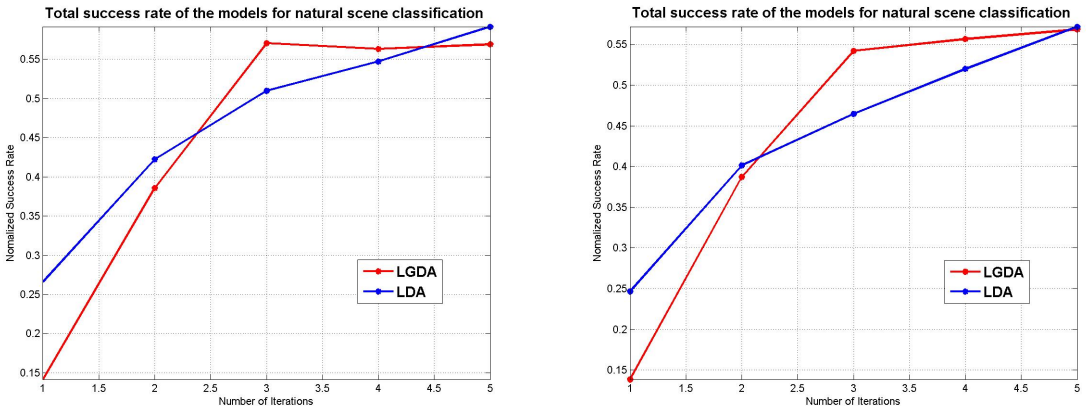


Figure 4-20: Comparison of the success rate of the online LGDA model against online LDA model for the natural scene classification, for 20 training image per step, for two different extracted number of topics.

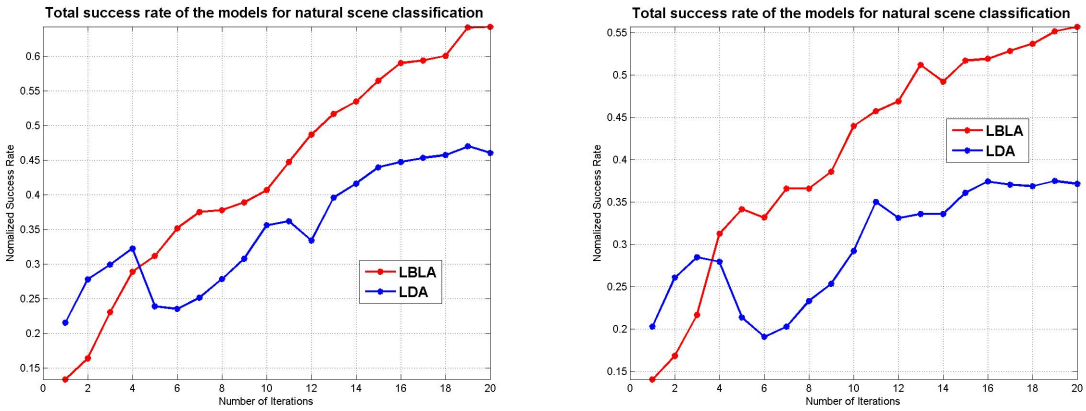


Figure 4-21: Sample two instances of the progression of the LBLA model success rate versus the LDA.

	Coast	Forest	Highway	Inside of cities	Opencountry	Streets	Tall building
Coast	216	1	86	3	50	3	2
Forest	3	242	15	53	4	51	0
Highway	40	1	69	5	6	3	15
Inside of cities	0	2	4	146	2	12	6
Open country	90	39	23	22	331	14	50
Streets	5	41	60	57	9	203	2
Tall building	6	2	3	22	8	6	281

Table 4.12: Optimal confusion matrix of the online LBLA model applied for the scenes classification task.

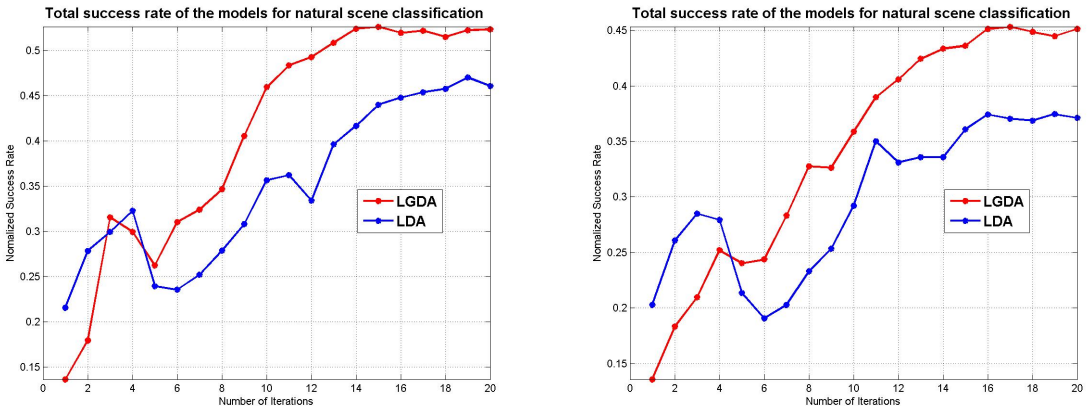


Figure 4-22: Sample two instances of the progression of the LGDA model success rate versus the LDA.

	Coast	**Forest**	**Highway**	**Inside of cities**	**Opencountry**	**Streets**	**Tall building**
Coast	282	4	130	5	98	7	14
Forest	9	287	24	84	24	167	3
Highway	19	2	55	13	16	6	94
Inside of cities	4	19	23	157	1	58	9
Open country	38	7	19	24	241	19	66
Streets	8	7	8	18	30	32	13
Tall building	0	2	1	7	0	3	157

Table 4.13: Optimal confusion matrix of the online LGDA model applied for the scenes classification task.

Chapter 5

Conclusion and future work

In this thesis we proposed several machine learning algorithms for different real world applications. We thoroughly analyzed the inherent drawbacks of the current models and proposed adequate theoretical improvements that resulted in new models with improved total performance.

In chapter 2, we focused a large part of our thesis on developing hierarchical models that capture the hierarchical nature of the available information more realistically. For that aim we proposed new hierarchical models based on the Dirichlet, generalized Dirichlet and Beta-Liouville distributions. Based on the performed experiments on the models in comparison with the existing hierarchical, our models, show superior performance. The emphasis of the proposed models is on single topic data classification.

In chapter 3, we proposed a new adaptable general learning hierarchical model. The model is based on the visual words approach. It was shown in the experimental results that the proposed model shows substantial improvement in hierarchical classification accuracy in comparison to the static models proposed in chapter 2. The improvement is achieved through applying several saliency factors in the learning process. In addition to that the learning algorithm proposed in this chapter allows our model to expand beyond the static hierarchical structures. The model proves

efficiency while dealing with unknown classes and as observed in the experiments succeeds in deciding the location of the new class within the hierarchy quite efficiently.

In chapter 4, we focused our research on multi-topic models. We proposed two models to improve the accuracy of existing multi-topic models. We considered a benchmark model LDA [12] and we developed two distinct multi-topic models based on it. Our first model considered the generalized Dirichlet assumption and the second was based on the Beta-Liouville assumption. We showed that our two models contain the LDA model as their special case but offer a more versatile character facing different data models. We tested the models and compared them with LDA for 3 different applications, text classication, natural scene classication and action recognition in video sequences. The models in all applications show similar or improved results in comparison to the benchmark. Later in chapter 4 we extended our models for online learning and we again showed the superiority of our models performances in comparison to the benchmark model.

We believe that the perspective of future work in this field is quite extensive. One trend that could be followed is developing fully automatic learning hierarchical structures that can overcome the structural restrictions of our proposed models. Another research possibility is looking for better priors for data modeling. The multi-topic models we proposed already are fully capable of being adapted to hierarchical learning models. One could imagine that the combination of the successful multi-topic models with hierarchical learning models can lead to successful learning models.

Appendix A

Appendixes

A.1 Appendix 1: Relationship between Parent and children nodes in hierarchical generalized Dirichlet model

By the definition of the model we know that for all the nodes, except for the root node, we have:

$$\vec{\theta_I} \sim GD(f(\vec{\theta}_{pa(I)}), g(\vec{\theta}_{pa(I)})) \tag{A.1}$$

Also from the hierarchical condition for the model we have:

$$E[\vec{\theta}_I|\theta_{pa(I)}] = \vec{\theta}_{pa(I)} \tag{A.2}$$

Using the formula for the average of the generalized Dirichlet distribution we have:

$$\theta_{pa(I)}(1) = \frac{f_I(1)}{f_I(1) + g_I(1)} \longrightarrow$$
$$f_I(1) = \frac{\theta_{pa(I)}(1)}{1 - \theta_{pa(I)}(1)} g_I(1)$$

For the second element of the hierarchical generalized Dirichlet vector and using the above equations we have:

$$\theta_{pa(I)}(2) = \frac{f_I(2)}{f_I(2)+g_I(2)} \frac{1}{\frac{f_I(1)}{g_I(1)}+1} \longrightarrow$$
$$\theta_{pa(I)}(2) = \frac{f_I(2)}{f_I(2)+g_I(2)} \frac{1}{\frac{\theta_{pa(I)}(1)}{1-\theta_{pa(I)}(1)}+1} \longrightarrow$$
$$g_I(2) = \frac{1-\sum_{k=1}^{2}\theta_{pa}(k)}{\theta_{pa}(2)} f_I(2)$$

Assuming that the relationship holds for all the parameters until the $(k-1)-th$, for the $k-th$ parameter we have:

$$\theta_{pa(i)}(k) = \frac{f_i(k)}{f_i(k)+g_i(k)} \frac{1}{\frac{\theta_{pa(i)}(1)}{1-\theta_{pa(i)}(1)}+1} \cdots$$
$$\frac{1}{\frac{\theta_{pa(i)}(k-1)}{1-\sum_{kk=1}^{k-1}\theta_{pa(i)}(kk)}+1} \longrightarrow$$

By simple mathematical algebra on each of the mathematical fractions of the above equation we end up having:

$$\theta_{pa(i)}(k) = \frac{f_i(k)}{f_i(k)+g_i(k)} \frac{1-\theta_{pa(i)}(1)}{1} \times$$
$$\frac{1-\sum_{kk=1}^{2}\theta_{pa(i)}(kk)}{1-\theta_{pa(i)}(1)} \cdots \frac{1-\sum_{kk=1}^{k-1}\theta_{pa(i)}(kk)}{1-\sum_{kk=1}^{k-2}\theta_{pa(i)}(kk)}$$

As it can be seen in the above equations the nominator of one fraction is cancelled with the denominator of the next one and so eventually the entire relationship is reduced to:

$$\theta_{pa(i)}(k) = \frac{f_i(k)}{f_i(k)+g_i(k)}(1-\sum_{kk=1}^{k-1}\theta_{pa(i)}(kk))$$

The same replacement as for the first parameter thus leads to the general parameter relationship equation:

$$g_{pa(i)}(k) = \frac{(1 - \sum_{kk=1}^{k} \theta_{pa(i)}(kk)}{\theta_{pa(i)}(k)} f_{pa(i)}(K)$$

A.2 Appendix 2: Relationship between hierarchical generalized Dirichlet and hierarchical Dirichlet models

It can be seen from the probability density function of the generalized Dirichlet distribution $GD(\vec{\alpha}, \vec{\beta})$ that under the following condition the generalized Dirichlet distribution is reduced to the Dirichlet distribution:

$$\beta(l) = \alpha(l+1) + \beta(l+1)$$

for the hierarchical generalized Dirichlet model therefore the reduction conditions will be:

$$g_I(l) = f_I(l+1) + g_I(l+1)$$

from the relationship derived in Appendix a, we have:

$$\frac{(1 - \sum_{k=1}^{l} \theta_{pa(I)}(k)}{\theta_{pa(I)}(l)} f_I(l) = f_I(l+1) + \frac{(1 - \sum_{k=1}^{l+1} \theta_{pa(I)}(k)}{\theta_{pa(I)}(l+1)} f_I(l+1)$$

Rearranging the above equation leads to the following restrain for preserving the Dirichlet condition:

$$\frac{f_I(l)}{\theta_{pa(I)(l)}} = \frac{f_I(l+1)}{\theta_{pa(I)(l+1)}}$$

The above relationship only holds when the value of the both sides equals a constant value σ independent of the values of $f_I(l)$ and $g_I(l)$.
Therefore for the reduction to hierarchical Dirichlet model we have:

$$f_I(l) = \sigma \theta_{pa(I)}(l)$$

A.3 Appendix 3: Exponential Form of the Generalized Dirichlet Distribution

In this Appendix we bring the exponential form of the generalized Dirichlet distribution [17] . The exponential form delivers us certain relationships necessary for obtaining the variational Bayes formulation.
generalized Dirichlet distribution belongs to the exponential family of distributions and therefore in general it can be presented as follows:

$$P(\vec{\theta}|\vec{\xi}) = Z_t(\vec{\theta}) \times \exp[\sum_{l=1}^{2d} G_l(\vec{\theta}) T_l(\vec{\theta})] \tag{A.3}$$

In above we have:

$$Z_t(\vec{\theta}) = \prod_{l=1}^{d} \frac{\Gamma(\alpha_l + \beta_l)}{\Gamma(\alpha_l) \times \Gamma(\beta_l)} \tag{A.4}$$

$$G_l(\vec{\xi}) = \alpha_l, l : 1, ..., d \tag{A.5}$$

$$G_l(\vec{\xi}) = \beta_{l-d} - \alpha_{l-d+1} - \beta_{l-d+1}, l = d+1, ..., 2d-1 \tag{A.6}$$

$$G_{2d}(\vec{\xi}) = \beta_d \tag{A.7}$$

$$T_l(\vec{\theta}) = \log(\theta_l), l = 1, ..., d \tag{A.8}$$

$$T_l(\vec{\theta}) = \log(1 - \sum_{t=1}^{l-d} \theta_t), l = d+1, ..., 2d \tag{A.9}$$

In the above $Z(\vec{\theta})$ is the normalization factor, $\vec{G}(\vec{\theta})$ is the natural parameter and $\vec{T}(\vec{\theta})$ is the sufficient statistics of the distribution. For the exponential family we know that the derivative of the logarithm of normalization factor with respect to the natural parameters equals the expected value of the sufficient statistics. Therefore we have:

$$E[\log(\theta_l)] = \Psi(\alpha_l + \beta_l) - \Psi(\alpha_l) - \Psi(\beta_l), l = 1, ..., d \tag{A.10}$$

$$E[\log(1 - \sum_{t=1}^{l} \theta_t)] = \Psi(\beta_l) - \Psi(\alpha_l + \beta_l), l = 1, ..., d \tag{A.11}$$

A.4 Appendix 4: Break down of the L parameter for LGDA

By factoring 4.33 we have:

$$L(\vec{\xi_q}, \Phi_{\mathbf{w}}; \vec{\xi}, \beta_w) = \\ E_q[\log\ p(\vec{\theta}|\vec{\xi})] + E_q[\log\ p(\mathbf{z})] + E_q[\log\ p(\mathbf{w}|\mathbf{z}, \beta_w)] \\ -E_q[\log\ q(\vec{\theta})] - E_q[\log\ q(\mathbf{z})] \tag{A.12}$$

We proceed with deriving each of the five factors of the above equation in the following.

$$E_q[\log p(\vec{\theta}|\xi)] = \\ \sum_{l=1}^{d}[\log\Gamma(\alpha_l + \beta_l) - \log\Gamma(\alpha_l) - \log\Gamma(\beta_l)] + \\ \sum_{l=1}^{d}[(\Psi(\gamma_l) - \Psi(\gamma_l + \delta_l))\alpha_l + \\ (\Psi(\delta_l) - \Psi(\gamma_l + \delta_l))(\beta_l - \alpha_{l+1} - \beta_{l+1}))] \tag{A.13}$$

$$E_q(\log\ p(\mathbf{z}|\vec{\theta})) = \sum_{n=1}^{N}\sum_{l=1}^{d}\phi_{nl}(\Psi(\gamma_l) - \Psi(\gamma_l + \delta_l)) + \\ \sum_{n=1}^{N}\phi_{n(d+1)}(\Psi(\delta_d) - \Psi(\gamma_d + \delta_d)) \tag{A.14}$$

$$E_q[\log\ p(\mathbf{w}|\mathbf{z}, \beta_w)] = \sum_{n=1}^{N}\sum_{l=1}^{d+1}\sum_{j=1}^{V}\phi_{nl}w_n^j\log(\beta_{w(lj)}) \tag{A.15}$$

In above $\beta_{w(lj)} = p(w_n^j = 1|z^l = 1)$.

$$E_q[\log\ q(\vec{\theta})] = \sum_{l=1}^{d}(\log\ \Gamma(\gamma_l + \delta_l) - \log\ \Gamma(\gamma_l) - \log\ \Gamma(\delta_l)) +$$

$$\sum_{l=1}^{d}[(\Psi(\gamma_l) - \Psi(\gamma_l + \delta_l))\gamma_l + (\Psi(\delta_l) - \Psi(\gamma_l + \delta_l))(\delta_l - \gamma_{l+1} - \delta_{l+1})] \tag{A.16}$$

$$E_q[\log\ q(\mathbf{z})] = \sum_{n=1}^{N}\sum_{l=1}^{d=1}\phi_{nl}\log(\phi_{nl}) \tag{A.17}$$

Having the above formulas we proceed with finding parameter estimation.

A.4.1 Variational Multinomial

In order to derive the parameter ϕ_{nl}, the probability that the nth word is generated by the l-th hidden topic, we proceed with maximizing A.45 with respect to ϕ_{nl}. Firstly we separate the terms A.45 containing ϕ_{nl}:

$$L[\phi_{nl}] = \phi_{nl}(\Psi(\gamma_l) - \Psi(\gamma_l + \delta_l)) + \phi_{nl}\log\ \beta_{w(lv)} - \phi_{nl}\log\ \phi_{nl} + \lambda_n(\sum_{ll=1}^{d+1}\phi_{n(ll)} - 1) \tag{A.18}$$

and

$$L[\phi_{n(d+1)}] = \phi_{n(d+1)}(\Psi(\delta_d) - \Psi(\gamma_d + \delta_d)) + \phi_{n(d+1)}\log\ \beta_{(d+1)v} - \phi_{n(d+1)}\log\ \phi_{n(d+1)} +$$
$$\lambda_n(\sum_{ll=1}^{d+1}\phi_{n(ll)} - 1) \tag{A.19}$$

and therefore we have:

$$\frac{\partial L}{\partial \phi_{nl}} = (\Psi(\gamma_l) - \Psi(\gamma_l + \delta_l)) + \log\ \beta_{lv} - \log\ \phi_{nl} - 1 + \lambda_n \tag{A.20}$$

and

$$\frac{\partial L}{\partial \phi_{n(d+1)}} = (\Psi(\delta_d) - \Psi(\gamma_d + \delta_d)) + \log\ \beta_{(d+1)v} - \log\ \phi_{n(d+1)} - 1 + \lambda_n \quad \text{(A.21)}$$

setting the above equation to zero leads to

$$\phi_{nl} = \beta_{lv} e^{(\lambda_n - 1)} e^{(\Psi(\gamma_l) - \Psi(\gamma_l + \delta_l))} \quad \text{(A.22)}$$

$$\phi_{n(d+1)} = \beta_{(d+1)v} e^{(\lambda_n - 1)} e^{(\Psi(\delta_d) - \Psi(\gamma_d + \delta_d))} \quad \text{(A.23)}$$

considering that $\sum_{ll=1}^{d+1} \phi_{n(ll)} = 1$ for the normalization factor we have:

$$e^{\lambda_n - 1} = \frac{1}{\sum_{l=1}^{d} \beta_{lv} e^{(\Psi(\gamma_l) - \Psi(\gamma_l + \delta_l))} + \beta_{(d+1)v} e^{(\Psi(\delta_d) - \Psi(\gamma_d + \delta_d))}} \quad \text{(A.24)}$$

A.4.2 Variational generalized Dirichlet

To find the update equations for the variational generalized Dirichlet we again proceed with separating the terms in A.45 containing the variation generalized Dirichlet parameters.

$$L[\vec{\xi}_q] = \sum_{l=1}^{d}[(\Psi(\gamma_l) - \Psi(\gamma_l + \delta_l))\alpha_l + (\Psi(\delta_l) - \Psi(\gamma_l + \delta_l)) \times (\beta_l - \alpha_{l+1} - \beta_{l+1})] + \sum_{n=1}^{N} \phi_{nl}(\Psi(\gamma_l) - \Psi(\gamma_l + \delta_l)) +$$
$$\sum_{n=1}^{N} \phi_{n(d+1)}(\Psi(\gamma_d) - \Psi(\gamma_d + \delta_d)) - [\sum_{l=1}^{d}(\log\ \Gamma(\gamma_l + \delta_l) - \log\ \Gamma(\gamma_l) - \log\ \Gamma(\delta_l)) + \sum_{l=1}^{d}[(\Psi(\gamma_l) - \Psi(\gamma_l + \delta_l))\gamma_l +$$
$$(\Psi(\delta_l) - \Psi(\gamma_l + \delta_l))(\delta_l - \gamma_{l+1} - \delta_{l+1})]] \quad \text{(A.25)}$$

Setting the derivative of the above equation to zero, leads to the following update equations:

$$\gamma_l = \alpha_l + \sum_{n=1}^{N} \phi_{nl} \quad \text{(A.26)}$$

$$\delta_l = \beta_l + \sum_{n=1}^{N} \sum_{ll=l+1}^{d+1} \phi_{n(ll)} \quad \text{(A.27)}$$

A.4.3 Topic based multinomial

In this appendix we derive the update equations necessary for estimating β_w. Maximizing A.45 with of the β_w leads to the same equation set as of LDA and we have:

$$L[\beta_w] = \sum_{d=1}^{M}\sum_{n=1}^{N_s}\sum_{l=1}^{k+1}\sum_{j=1}^{V}\phi_{dnl}w_{dn}^{j}\log\beta_{w(lj)} + \sum_{l=1}^{k+1}\lambda_l\left(\sum_{j=1}^{V}\beta_{w(ij)} - 1\right) \tag{A.28}$$

Taking the derivative with respect to $\beta_{w(lj)}$ and setting it to zero gives:

$$\beta_{w(lj)} \propto \sum_{d=1}^{M}\sum_{n=1}^{N_d}\phi_{dnl}w_{dn}^{j} \tag{A.29}$$

A.4.4 Generalized Dirichlet parameters

We choose the terms of equation A.45 containing the generalized Dirichlet parameters $\vec{\xi}$.

$$L[\vec{\xi}] = \sum_{m=1}^{M}(\log(\Gamma(\alpha_l+\beta_l)) - \log(\Gamma(\alpha_l)) - \log(\Gamma(\beta_l))) + \sum_{m=1}^{M}[(\Psi(\gamma_{ml}) - \Psi(\gamma_{ml}+\delta_{ml}))\alpha_l + (\Psi(\delta_{ml}) - \Psi(\gamma_{ml}+\delta_{ml}))\beta_l] \tag{A.30}$$

The derivative of the above equation in respect to the generalized Dirichlet parameters gives:

$$\frac{\partial L[\vec{\xi}]}{\partial\alpha_l} = M(\Psi(\alpha_l+\beta_l) - \Psi(\alpha_l)) + \sum_{m=1}^{M}(\Psi(\gamma_{ml}) - \Psi(\gamma_{ml}+\delta_{ml})) \tag{A.31}$$

and

$$\frac{\partial L[\vec{\xi}]}{\partial\beta_l} = M(\Psi(\alpha_l+\beta_l) - \Psi(\beta_l)) + \sum_{m=1}^{M}(\Psi(\delta_{ml}) - \Psi(\gamma_{ml}+\delta_{ml})) \tag{A.32}$$

It can be seen from the equations above that the derivative of A.45 with respect to each of the generalized Dirichlet parameters α_l and β_l depend not only on their own values but also on each other. To solve the optimization problem therefore we propose

using the Newton-Raphson method. In order to solve the Newton Raphson method we need to have access to the Hessian Matrix of A.45 in respect to the parameter space. The Hessian matrix of the likelihood however takes a peculiarly interesting form as follows:

$$\frac{\partial^2 L[\vec{\xi}]}{\partial \alpha_l^2} = M(\Psi`(\alpha_l + \beta_l) - \Psi`(\alpha_l)) \tag{A.33}$$

$$\frac{\partial^2 L[\vec{\xi}]}{\partial \beta_l^2} = M(\Psi`(\alpha_l + \beta_l) - \Psi`(\beta_l)) \tag{A.34}$$

$$\frac{\partial^2 L[\vec{\xi}]}{\partial \alpha_l \partial \beta_l} = M(\Psi`(\alpha_l + \beta_l)) \tag{A.35}$$

$$\frac{\partial^2 L[\vec{\xi}]}{\partial \beta_l \partial \alpha_l} = M(\Psi`(\alpha_l + \beta_l)) \tag{A.36}$$

The other entries of the Hessian Matrix are zero. The above equations give the Hessian matrix a block diagonal form and therefore the reverse Hessian matrix will be the reverse of 2×2 matrix on the diagonal and is easily derived.

A.5 Appendix 5: Exponential Form of the Beta-Liouville Distribution

Here, we present the exponential form of the Beta-Liouville distribution. The exponential form delivers us certain relationships necessary for developing the variational Bayes inference that we shall adopt. It is straightforward to show the Beta-Liouville can be written in the following exponential form [17]:

$$P(\vec{\theta}|\vec{\xi}) = \exp[\sum_{l=1}^{D+2} G_l(\vec{\theta}) T_l(\vec{\theta}) - \Phi(\vec{\xi})] \tag{A.37}$$

In above we have:

$$G_d(\vec{\xi}) = \alpha_d, d = 1, \ldots, D \quad G_{D+1} = (\vec{\xi}) = \alpha \quad G_{D+2} = \beta \tag{A.38}$$

$$T_d(\vec{\theta}) = \log(\theta_d) - \log(\sum_{l=1}^{D} \theta_l), d = 1, \ldots, D \quad T_{D+1}(\vec{\theta}) = \log(\sum_{l=1}^{D} \theta_l) \quad T_{D+2}(\vec{\theta}) = \log(1 - \sum_{l=1}^{D} \theta_l) \tag{A.39}$$

$$-\Phi(\vec{\xi}) = \log(\Gamma(\alpha)) + \log(\Gamma(\beta)) + \sum_{l=1}^{D} \log(\Gamma(\alpha_l)) - \log(\Gamma(\sum_{l=1}^{D} \alpha_l)) - \log(\Gamma(\alpha + \beta)) \tag{A.40}$$

In the above $-\Phi(\vec{\xi})$ is the log normalization factor, $\vec{G}(\vec{\xi}) = (G_1(\vec{\xi}), \ldots, G_{2D}(\vec{\xi}))$ is the natural parameter and $\vec{T}(\vec{\theta}) = (T_1(\vec{\theta}), \ldots, T_{2D}(\vec{\theta}))$ is the sufficient statistics of the distribution. For the exponential family of distributions, we know that the derivative of the logarithm of normalization factor with respect to the natural parameters equals the expected value of the sufficient statistics. Therefore, we have:

$$E_\theta[\log \theta_d - \log(\sum_{l=1}^{D} \theta_l)] = \Psi(\alpha_d) - \Psi(\sum_{l=1}^{D} \alpha_l) \tag{A.41}$$

$$E_\theta[\log(\sum_{l=1}^{D} \theta_l)] = \Psi(\alpha) - \Psi(\alpha + \beta) \tag{A.42}$$

$$E_\theta[\log(1 - \sum_{l=1}^{D} \theta_l)] = \Psi(\beta) - \Psi(\alpha + \beta) \tag{A.43}$$

from the above equations we have:

$$E_\theta[\log \theta_d] = \Psi(\alpha_d) - \Psi(\sum_{d=1}^{D} \alpha_d) + \Psi(\alpha) - \Psi(\alpha + \beta) \tag{A.44}$$

The above equations allows us to derive the needed variational equations.

A.6 Break down of the L parameter for LBLA

By factorizing $L(\vec{\xi_q}, \Phi_{\boldsymbol{w}}; \vec{\xi}, \beta_w)$ in Eq. 4.33, we obtain:

$$L(\vec{\xi_q}, \Phi_{\boldsymbol{w}}; \vec{\xi}, \beta_w) = E_q[\log p(\vec{\theta}|\vec{\xi})] + E_q[\log p(\boldsymbol{z})] + E_q[\log p(\boldsymbol{w}|\boldsymbol{z}, \beta_w)] - E_q[\log q(\vec{\theta})] - E_q[\log q(\boldsymbol{z})] \tag{A.45}$$

We proceed with deriving each of the five factors of the above equation in the following.

$$\begin{aligned} E_q[\log p(\vec{\theta}|\xi)] &= \log(\Gamma(\sum_{d=1}^{D} \alpha_d)) + \log(\Gamma(\alpha + \beta)) - \log(\Gamma(\alpha)) \\ &\quad - \log(\Gamma(\beta)) - \sum_{d=1}^{D} \log \Gamma(\alpha_d) + \sum_{d=1}^{D} \alpha_d (\Psi(\gamma_d) - \Psi(\sum_{l=1}^{D} \gamma_l)) \\ &+ \alpha(\Psi(\alpha_\gamma) - \Psi(\alpha_\gamma + \beta_\gamma)) + \beta(\Psi(\beta_\gamma) - \Psi(\alpha_\gamma + \beta_\gamma)) + \beta(\Psi(\beta_\gamma) - \Psi(\alpha_\gamma + \beta_\gamma)) \end{aligned} \tag{A.46}$$

and

$$\begin{aligned} E_q(\log p(\boldsymbol{z}|\vec{\theta})) &= \sum_{n=1}^{N} \sum_{d=1}^{D} \phi_{nd} (\Psi(\gamma_d) - \Psi(\sum_{l=1}^{D} \gamma_l) + \Psi(\alpha_\gamma) - \Psi(\alpha_\gamma + \beta_\gamma)) \\ &+ \sum_{n=1}^{N} \phi_{n(D+1)} (\Psi(\beta_\gamma) - \Psi(\alpha_\gamma + \beta_\gamma)) \end{aligned} \tag{A.47}$$

$$E_q[\log p(\boldsymbol{w}|\boldsymbol{z}, \beta_w)] = \sum_{n=1}^{N} \sum_{l=1}^{D+1} \sum_{j=1}^{V} \phi_{nl} w_n^j \log(\beta_{w(lj)}) \tag{A.48}$$

where $\beta_{w(lj)} = p(w_n^j = 1|z^l = 1)$.

$$\begin{aligned} E_q[\log\ q(\vec{\theta})] &= \log(\Gamma(\sum_{l=1}^{D}\gamma_l)) + \log(\Gamma(\alpha_\gamma+\beta_\gamma)) - \log(\Gamma(\alpha_\gamma)) - \log(\gamma(\beta_\gamma)) - \sum_{l=1}^{D}\log\Gamma(\gamma_l) \\ &+ \sum_{d=1}^{D}\gamma_d(\Psi(\gamma_d) - \Psi(\sum_{dd=1}^{D}\gamma_{dd}) + \Psi(\alpha_\gamma) - \Psi(\alpha_\gamma+\beta_\gamma)) \\ &+ \alpha_\gamma(\Psi(\alpha_\gamma) - \Psi(\alpha_\gamma) - \Psi(\alpha_\gamma+\beta_\gamma)) + \beta_\gamma(\Psi(\beta_\gamma) - \Psi(\alpha_\gamma+\beta_\gamma)) \end{aligned} \tag{A.49}$$

$$E_q[\log q(\boldsymbol{z})] = \sum_{n=1}^{N}\sum_{l=1}^{D+1}\phi_{nl}\log(\phi_{nl}) \tag{A.50}$$

Having the above formulas we proceed with finding the parameters estimates.

Appendix 2.1: Variational Multinomial

In order to derive the parameter ϕ_{nl}, the probability that the nth word is generated by the l-th hidden topic, we proceed with maximizing A.45 with respect to ϕ_{nl}. Firstly, we separate the terms A.45 containing ϕ_{nl}:

$$L[\phi_{nl}] = \phi_{ni}(\Psi(\gamma_i) - \Psi(\sum_{l=1}^{D}\gamma_l)) + \phi_{ni}\log\beta_{w(iv)} - \phi_{ni}\log\phi_{ni} + \lambda_n(\sum_{l=1}^{D}\phi_{n(l)} - 1)$$

and

$$\begin{aligned} L[\phi_{n(D+1)}] = \phi_{n(D+1)}(\Psi(\beta_\gamma - \Psi(\alpha_\gamma+\beta_\gamma)) + \phi_{n(D+1)}\log\beta_{(D+1)v} - \phi_{n(D+1)}\log\phi_{n(D+1)} \\ +\lambda_n(\sum_{i=1}^{D}\phi_{n(i)} - 1) \end{aligned} \tag{A.51}$$

and therefore we have:

$$\frac{\partial L}{\partial\phi_{nl}} = (\Psi(\gamma_d) - \Psi(\sum_{l=1}^{D}\gamma_l)) + \log\beta_{w(iv)} - \log\phi_{ni} - 1 + \lambda_n \tag{A.52}$$

and

$$\frac{\partial L}{\partial \phi_{n(D+1)}} = (\Psi(\beta_\gamma) - \Psi(\alpha_\gamma + \beta_\gamma)) + \log \beta_{(D+1)v} - log\phi_{n(D+1)} - 1 + \lambda_n \quad \text{(A.53)}$$

setting the above equation to zero leads to

$$\phi_{nl} = \beta_{lv} e^{(\lambda_n - 1)} e^{(\Psi(\gamma_i) - \Psi(\sum_{i=1}^{D} \gamma_i))} \quad \text{(A.54)}$$

$$\phi_{n(D+1)} = \beta_{(D+1)v} e^{(\lambda_n - 1)} e^{(\Psi(\beta_\gamma) - \Psi(\alpha_\gamma + \beta_\gamma))} \quad \text{(A.55)}$$

considering that $\sum_{d=1}^{D+1} \phi_{n(d)} = 1$ for the normalization factor we have:

$$e^{\lambda_n - 1} = \frac{1}{\beta_{(D+1)v} e^{(\Psi(\beta_\gamma) - \Psi(\alpha_\gamma + \beta_\gamma))} + \sum_{i=1}^{D} \beta_{iv} e^{(\Psi(\gamma_d) - \Psi(\sum_{i=1}^{D} \gamma_i))}} \quad \text{(A.56)}$$

Appendix 2.2: Variational Beta-Liouville

To find the update equations for the variational BL we again proceed with separating the terms in A.45 containing the variation BL parameters.

$$\begin{aligned} & L[\vec{\xi_q}] = \alpha_d(\Psi(\gamma_d)) - \Psi(\sum_{l=1}^{D} \gamma_l) + \alpha(\Psi(\alpha_\gamma) - \Psi(\alpha_\gamma + \beta_\gamma)) + \beta(\Psi(\beta_\gamma) - \Psi(\alpha_\gamma + \beta_\beta)) + \\ & \sum_{n=1}^{N} \phi_{ni}(\Psi(\gamma_i) - \psi(\sum_{l=1}^{D} \gamma_l) + \Psi(\alpha_\gamma) - \Psi(\alpha_\gamma + \beta_\gamma)) + \sum_{n=1}^{N} \phi_{n(D+1)}(\Psi(\beta_\gamma) - \Psi(\alpha_\gamma + \beta_\gamma)) - \\ & (\log(\Gamma(\sum_{l=1}^{D} \gamma_l)) + \log(\gamma(\alpha_\gamma + \beta_\gamma)) - \log(\Gamma(\alpha_\gamma)) - \log(\gamma(\beta_\gamma)) - \log \gamma(\gamma_i) \\ + \quad & \gamma_i(\Psi(\gamma_i) + \Psi(\alpha_\gamma) - \Psi(\alpha_\gamma + \beta_\gamma)) - \Psi(\sum_{l=1}^{D} \gamma_l) \sum_{d=1}^{D} \gamma_d + \alpha_\gamma(\Psi(\alpha_\gamma) - \Psi(\alpha_\gamma + \beta_\gamma)) \\ + \quad & \beta_\gamma(\Psi(\beta_\gamma) - \Psi(\alpha_\gamma + \beta_\gamma))) \end{aligned} \quad \text{(A.57)}$$

selecting the terms containing variational BL variables $\gamma_i, \alpha_\gamma, \beta_\gamma$ we have:

$$\begin{aligned} l(\gamma_i) &= \alpha_i(\Psi(\gamma_i)) - \Psi(\sum_{l=1}^{D}\gamma_l)\sum_{l=1}^{D}\alpha_l + \sum_{n=1}^{N}\phi_{ni}(\Psi(\gamma_i) - \Psi(\sum_{l=1}^{D}\gamma_l)) \\ &- (\log(\Gamma(\sum_{l=1}^{D}\gamma_l)) - \log\Gamma(\gamma_i) + \gamma_i(\Psi(\gamma_i)) - \Psi(\sum_{l=1}^{D}\gamma_l)\sum_{d=1}^{D}\gamma_d) \end{aligned} \tag{A.58}$$

$$\begin{aligned} L[\alpha_\gamma] &= \alpha(\Psi(\alpha_\gamma) - \Psi(\alpha_\gamma+\beta_\gamma)) + \beta(-\Psi(\alpha_\gamma+\beta_\gamma)) + (\Psi(\alpha_\gamma) - \Psi(\alpha_\gamma+\beta_\gamma))\sum_{n=1}^{N}\sum_{i=1}^{D}\phi_{ni} \\ &+ \sum_{n=1}^{N}\phi_{n(D+1)}(-\Psi(\alpha_\gamma+\beta_\gamma)) \\ &- (\log(\alpha_\gamma+\beta_\gamma) - \log(\Gamma(\alpha_\gamma)) + \alpha_\gamma(\Psi(\alpha_\gamma) - \Psi(\alpha_\gamma+\beta_\gamma) + \beta_\gamma(-\Psi(\alpha_\gamma+\beta_\gamma)) \end{aligned} \tag{A.59}$$

Taking the derivative of the above equations in respect to their BL parameter gives:

$$\begin{aligned} \frac{\partial L[\gamma_i]}{\partial\gamma_i} &= \alpha_i\Psi'(\gamma_i) - \Psi'(\sum_{l=1}^{D}\gamma_l)\sum_{l=1}^{D}\alpha_l + \Psi'(\gamma_i)\sum_{n=1}^{N}\phi_{ni} - D\Psi'(\sum_{l=1}^{D}\gamma_l)\sum_{n=1}^{N}\phi_{ni} \\ &- (\Psi(\sum_{l=1}^{D}\gamma_l) + \gamma_i\Psi'(\gamma_i) - \Psi'(\sum_{l=1}^{D}\gamma_l)\sum_{d=1}^{D}\gamma_d - \Psi(\sum_{l=1}^{D}\gamma_l)) \end{aligned} \tag{A.60}$$

and

$$\begin{aligned} \frac{\partial L[\gamma_i]}{\partial\alpha_\gamma} &= \alpha(\Psi'(\alpha_\gamma) - \Psi'(\alpha_\gamma+\beta_\gamma)) + \beta(-\Psi'(\alpha_\gamma+\beta_\gamma)) + (\Psi'(\alpha_\gamma) - \Psi'(\alpha_\gamma+\beta_\gamma))\sum_{n=1}^{N}\sum_{d=1}^{D}\phi_{nd} \\ &+ \sum_{n=1}^{N}\phi_{n(D+1)}(-\Psi'(\alpha_\gamma+\beta_\gamma)) - (\alpha_\gamma(\Psi'(\alpha_\gamma) - \Psi'(\alpha_\gamma+\beta_\gamma)) + \beta_\gamma(-\Psi'(\alpha_\gamma+\beta_\gamma))) \end{aligned} \tag{A.61}$$

Setting the above equations to zero leads to the variational BL update equations.

$$\gamma_i = \alpha + \sum_{n=1}^{N} \phi_{ni} \tag{A.62}$$

$$\alpha_\gamma = \alpha + \sum_{n=1}^{N} \sum_{d=1}^{D} \phi_{nd} \tag{A.63}$$

$$\beta_\gamma = \beta + \sum_{n=1}^{N} \phi_{n(D+1)} \tag{A.64}$$

Appendix 2.3: Topic based multinomial

In this appendix we derive the updating equations necessary for estimating β_w. Maximizing Eq. A.45 with respect to β_w leads to the same equation as in the LDA case:

$$L[\beta_w] = \sum_{d=1}^{M} \sum_{n=1}^{N_s} \sum_{l=1}^{D+1} \sum_{j=1}^{V} \phi_{dnl} w_{dn}^{j} \log \beta_{w(lj)} + \sum_{l=1}^{D+1} \lambda_l (\sum_{j=1}^{V} \beta_{w(lj)} - 1) \tag{A.65}$$

Taking the derivative with respect to $\beta_{w(lj)}$ and setting it to zero gives:

$$\beta_{w(lj)} \propto \sum_{d=1}^{M} \sum_{n=1}^{N_d} \phi_{dnl} w_{dn}^{j} \tag{A.66}$$

Appendix 2.4: Beta-Liouville parameters

We choose the terms of A.45 containing the BL parameters $\vec{\xi}$.

$$L[\vec{\xi}] = \sum_{m=1}^{M} (\log(\Gamma(\sum_{l=1}^{D} \alpha_l)) + \log(\Gamma(\alpha+\beta)) - \log\Gamma(\alpha) - \log\gamma(\beta) - \sum_{i=1}^{D} \log\Gamma(\alpha_i)$$
$$+ \sum_{i=1}^{D} \alpha_i (\Psi(\gamma_{mi}) - \Psi(\sum_{l=1}^{D} \gamma_{m(l)})) + \alpha(\Psi(\alpha_{m\gamma}) - \Psi(\alpha_{m\gamma}+\beta_{m\gamma})) + \beta(\Psi(\beta_{m\gamma}) - \Psi(\alpha_{m\gamma}+\beta_{m\gamma})))$$

The derivative of the above equation in respect to the BL parameters gives:

$$\frac{\partial L[\vec{\xi}]}{\partial \alpha_l} = M(\Psi(\sum_{l=1}^{D} \alpha_l) - \Psi(\alpha_l)) + \sum_{m=1}^{M}(\Psi'(\gamma_{ml}) - \Psi(\sum_{l=1}^{D} \gamma_{m(l)})) \tag{A.67}$$

$$\frac{\partial L[\vec{\xi}]}{\partial \alpha} = M[\Psi(\alpha + \beta) - \Psi(\alpha)] + \sum_{m=1}^{M}(\Psi(\alpha_{m\gamma}) - \Psi(\alpha_{m\gamma} + \beta_{m\gamma})) \tag{A.68}$$

$$\frac{\partial L[\vec{\xi}]}{\partial \beta} = M[\Psi(\alpha + \beta) - \Psi(\beta)] + \sum_{m=1}^{M}(\Psi(\beta_{m\gamma}) - \Psi(\alpha_{m\gamma} + \beta_{m\gamma})) \tag{A.69}$$

It can be seen from the equations above that the derivative of A.45 with respect to each of the BL parameters α_l and β_l depend not only on their own values but also on each other. To solve the optimization problem therefore we use the Newton-Raphson method. In order to solve the Newton Raphson method we need to compute the Hessian matrix of A.45 with respect to the parameter space as follows:

$$\frac{\partial^2 L[\vec{\xi}]}{\partial \alpha_l \alpha_j} = M(-\delta(i,j)\Psi'(\alpha_i) + \Psi'(\sum_{l=1}^{D} \alpha_l)) \tag{A.70}$$

$$\frac{\partial^2 L[\vec{\xi}]}{\partial \alpha^2} = M(\Psi'(\alpha + \beta) - \Psi'(\alpha)) \tag{A.71}$$

$$\frac{\partial^2 L[\vec{\xi}]}{\partial \beta \partial \alpha} = M\Psi'(\alpha + \beta) \tag{A.72}$$

$$\frac{\partial^2 L[\vec{\xi}]}{\partial \beta^2} = M(\Psi'(\alpha + \beta) - \Psi'(\beta)) \tag{A.73}$$

$$\tag{A.74}$$

The above Hessian matrix closely resembles the Hessian matrix of the Dirichlet parameters in LDA model. In fact, the above matrix can be divided into two completely separate matrices consisting of α_d, α and β parameters. The parameter derivation of each of the two parts will be identical to the Newton-Raphson model offered in LDA.

Bibliography

[1] A. Vailaya, M. A. T. Figueiredo, A. K. Jain and H-J. Zhang. Image Classification for Content-Based Indexing. *IEEE Transactions on Image Processing*, 10(1):117–130, 2001.

[2] A. S. Bakhtiari and N. Bouguila. A hierarchical statistical model for object classification. In *Multimedia Signal Processing (MMSP), 2010 IEEE International Workshop on*, pages 493–498, 2010.

[3] A. S. Bakhtiari and N. Bouguila. An expandable hierarchical statistical framework for count data modeling and its application to object classification. In *Proceedings of the IEEE 23rd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 817–824. IEEE, 2011.

[4] A. S. Bakhtiari and N. Bouguila. A novel hierarchical statistical model for count data modeling and its application in image classification. In *In proceedings of International Conference on Neural Information Processing (ICONIP)*, pages 332–340, 2012.

[5] A. S. Bakhtiari and N. Bouguila. Semisupervised online learning of hierarchical structures for visual object classification. *Multimedia Tools and Applications*, pages 1–18, 2013.

[6] A. S. Bakhtiari and N. Bouguila. A latent topic model based on the beta-liouville distribution. *Neural Computing and Applications*, 2014.

[7] A. S. Bakhtiari and N. Bouguila. Online learning for two novel latent topic models. *ICT-Eurasia*, 2014. Accepted paper.

[8] A. S. Bakhtiari and N. Bouguila. A variational bayes model for count data learning and classication. *International Scientific Journal Engineering Applications of Artificial Intelligence*, 2014.

[9] T. Bayes. An essay towards solving a problem in the doctrine of chances. *Phil. Trans. of the Royal Soc. of London*, 53:370–418, 1763.

[10] D. Blei, T. Griffiths, M. Jordan, and J. Tenenbaum. Hierarchical topic models and the nested Chinese restaurant process. In *Advances in Neural Information Processing Systems*. MIT Press, 2004.

[11] D. M. Blei, Thomas L. Griffiths, Michael I. Jordan, and Joshua B. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 2003.

[12] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[13] A. F. Bobick and A. D. Wilson. A state-based approach to the representation and recognition of gesture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(12):1325–1337, 1997.

[14] A.F. Bobick and Y.A. Ivanov. Action recognition using probabilistic parsing. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 196–202. IEEE, 1998.

[15] N. Bouguila. Count data modeling and classification using finite mixtures of distributions. *IEEE Transactions on Neural Networks*, 22(2):186–198, 2011.

[16] N. Bouguila. A liouville-based approach for discrete data categorization. In Sergei O. Kuznetsov, Dominik Slezak, Daryl H. Hepting, and Boris Mirkin, editors, *RSFDGrC*, volume 6743 of *Lecture Notes in Computer Science*, pages 330–337. Springer, 2011.

[17] N. Bouguila. Hybrid generative/discriminative approaches for proportional data modeling and classification. *IEEE Transactions on Knowledge and Data Engineering*, 24(12):2184–2202, 2012.

[18] N. Bouguila and D. Ziou. High-dimensional unsupervised selection and estimation of a finite generalized Dirichlet mixture model based on minimum message length. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1716–1731, 2007.

[19] N. Bouguila and D. Ziou. unsupervised learning of a finite discrete mixture: Applications to texture modeling and image databases summarization. *Journal of Visual Communication and Image Representation*, 15(4):295–309, 2007.

[20] N. Bouguila and D. Ziou. A nonparametric bayesian learning model: Application to text and image categorization. In Thanaruk Theeramunkong, Boonserm Kijsirikul, Nick Cercone, and Tu Bao Ho, editors, *PAKDD*, volume 5476 of *Lecture Notes in Computer Science*, pages 463–474. Springer, 2009.

[21] T. Brants. Test data likelihood for plsa models. *Information Retrieval*, 8(2):181–196, 2005.

[22] K. L. Caballero, J. Barajas, and R. Akella. The generalized dirichlet distribution in enhanced topic detection. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 773–782, New York, NY, USA, 2012. ACM.

[23] G. Casella and E. I. George. Explaining the gibbs sampler. *The American Statistician*, 46(3):167–174, 1992.

[24] R. Connor and J. Mosimann. Concepts of independence for proportions with a generalization of the Dirichlet distribution. *Journal of the American Statistical Association*, 64(325):194–206, 1969.

[25] J. D. Courtney. Automatic video indexing via object motion analysis. *Pattern Recognition*, 30(4):607–625, 1997.

[26] R. T. Cox. Probability, Frequency and Reasonable Expectation. *American Journal of Physics*, 14(1):1–13, 1946.

[27] G. Csurka, C.R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[28] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.

[29] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B*, 39(1):1–38, 1977.

[30] E.Nowak, F.Jurie, and B.Triggs. Sampling strategies for bag-of-features image classification. In *Proceedings of ECCV*, pages 490–503. Springer, 2006.

[31] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 524–531. IEEE Press, 2005.

[32] L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *Proceedings of CVPR*, pages 524–531, 2005.

[33] D. J. Felleman and D. C. Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cereberal Cortex*, 1(1):1–a–47, January 1991.

[34] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis, Second Edition (Chapman & Hall/CRC Texts in Statistical Science)*. Chapman and Hall/CRC, 2 edition, 2003.

[35] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253, 2007.

[36] D. Grangier, F. Monay, and S. Bengio. A discriminative approach for the retrieval of images from text queries. In Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, editors, *ECML*, volume 4212 of *Lecture Notes in Computer Science*, pages 162–173. Springer, 2006.

[37] T. Griffiths and M. Steyvers. A probabilistic approach to semantic representation. In *Proceedings of the 24th annual conference of the cognitive science society*, pages 381–386. Cognitive Science Society, 2002.

[38] A.K. Gupta and D. Song. Generalized liouville distribution. *Computers Mathematics with Applications*, 32(2):103 – 109, 1996.

[39] J. A. Hartigan. *Clustering algorithms*. Wiley, 1975.

[40] H. W. Haussecker and D. J. Fleet. Computing optical flow with physical models of brightness variation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):661–673, 2001.

[41] M. D. Hoffman, D. M. Blei, and F. R. Bach. Online learning for latent dirichlet allocation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 856–864. Curran Associates, Inc., 2010.

[42] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1/2):177–196, 2001.

[43] B. K. P. Horn and B. G. Schunck. Determining optical flow, 1980.

[44] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning (ECML)*, pages 137–142. Springer, 1998.

[45] N. Jojic and Y. Caspi. Capturing image structure with probabilistic index maps. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages I–212–I–219 Vol.1. IEEE, 2004.

[46] M. I. Jordan, Z. Ghahramani, T. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

[47] L. G. Kraft. A device for quantizing, grouping, and coding amplitude modulated pulses. M.Sc. Thesis, Dept. of Electrical Engineering, MIT, Cambridge, Mass., 1949.

[48] S.Kotz K.T.Fang and K.W.Ng. *Symmetric Multivariate and Related Distributions*. Chapman and Hall, 1990.

[49] S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[50] R. Kwitt, N. Vasconcelos, and N. Rasiwasia. Scene recognition on the semantic manifold. In Andrew W. Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *ECCV (4)*, volume 7575 of *Lecture Notes in Computer Science*, pages 359–372. Springer, 2012.

[51] J. J. Lee. LIBPMK: A pyramid match toolkit. Technical report, MIT Computer Science and Artificial Intelligence Laboratory, April 2008.

[52] L. Li, H. Su, E. P. Xing, and L. Fei-Fei. Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1378–1386. Curran Associates, Inc., 2010.

[53] D.G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of ICCV*, volume 2, pages 1150–1157, 1999.

[54] R. E. Madsen, D. Kauchak, and C. Elkan. Modeling word burstiness using the Dirichlet distribution. In *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, pages 545–552, Bonn, Germany, 2005. ACM Press.

[55] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1615–1630, 2005.

[56] T. P. Minka. Estimating a Dirichlet distribution. Unpublished paper available at http://research.microsoft.com/∼minka/papers/dirichlet/, Microsoft Research (Cambridge, UK), 2007.

[57] N. Bouguila. Spatial Color Image Databases Summarization. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Volume 1*, pages 953–956, Honolulu, HI, USA, 2007.

[58] N. Bouguila. Clustering of Count Data Using Generalized Dirichlet Multinomial Distributions. *IEEE Transactions on Knowledge and Data Engineering*, 20(4):462–474, 2008.

[59] N. Bouguila and D. Ziou. A Powerful Finite Mixture Model Based on the Generalized Dirichlet Distribution: Unsupervised Learning and Applications. In *Proceedings of the 17th International Conference on Pattern Recognition, ICPR2004*, pages 280–283, 2004.

[60] N. Bouguila and D. Ziou. Dirichlet-Based Probability Model Applied to Human Skin Detection. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 521–524, 2004.

[61] N. Bouguila and D. Ziou. Improving content based image retrieval systems using finite multinomial Dirichlet mixture. In *Proceedings of the 14th IEEE Workshop on Machine Learning for Signal Processing*, pages 23–32, Sao Luis, Brazil, Oct. 2004.

[62] N. Bouguila and K. Daoudi. Learning Concepts from Visual Scenes using a Binary Probabilistic Model. In *Proceedings of the IEEE International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–5, Rio De Janeiro, Brazil, 2009.

[63] N. Bouguila and M. N. Ghimire. Discrete visual features modeling via leave-one-out likelihood estimation and applications. *Journal of Visual Communication and Image Representation*, 21(7):613–626, 2010.

[64] N. Bouguila and W. ElGuebaly. A Generative Model for Spatial Color Image Databases Categorization. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 821–824, Las Vegas, Nevada, USA, 2008.

[65] N. Bouguila and W. ElGuebaly. On Discrete Data Clustering. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD2008), LNCS 5012*, pages 503–510, Osaka, Japan, 2008. Springer.

[66] N. Bouguila and W. ElGuebaly. Discrete data clustering using finite mixture models. *Pattern Recognition*, 42(1):33–42, 2009.

[67] N. Bouguila, D. Ziou and J. Vaillancourt. Unsupervised Learning of a Finite Mixture Model Based on the Dirichlet Distribution and its Application. *IEEE Transactions on Image Processing*, 13(11):1533–1543, 2004.

[68] N. Bouguila, D. Ziou and R. I. Hammoud. On Bayesian Analysis of a Finite Generalized Dirichlet Mixture Via a Metropolis-within-Gibbs Sampling. *Pattern Analysis and Applications*, 12(2):151–166, 2009.

[69] Radford M. Neal. Probabilistic inference using markov chain monte carlo methods, 1993.

[70] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2):103–134, 2000.

[71] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[72] E. J. G. Pitman. Sufficient statistics and intrinsic accuracy. *Mathematical Proceedings of the Cambridge Philosophical Society*, 32(04):567–579, 1936.

[73] A. Popescul, L. H. Ungar, D. M. Pennock, and S. Lawrence. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence (UAI)*, pages 437–444. Morgan Kaufmann, 2001.

[74] S. Boutemedjet, D. Ziou and N. Bouguila. Unsupervised Feature Selection for Accurate Recommendation of High-Dimensional Image Data. In *Advances in Neural Information Processing Systems (NIPS)*, pages 177–184, 2007.

[75] C. Marlot S. Thorpe, D. Fize. Speed of processing in the human visual system. *Nature*, 381(5682):520–522, June 1996.

[76] R. Salakhutdinov and G. E. Hinton. Replicated softmax: an undirected topic model. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1607–1614. Curran Associates, Inc., 2009.

[77] F. Scalzo and J.H. Piater. Adaptive patch features for object class recognition with learned hierarchical models. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2007.

[78] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[79] J. Sivic, B. C. Russell, A. Zisserman, W. T. Freeman, and A. A. Efros. Unsupervised discovery of visual object class hierarchies. In *Proceedings of CVPR*, pages 1–8, 2008.

[80] M. Steyvers and T. Griffiths. Probabilistic topic models. In *Handbook of Latent Semantic Analysis*, pages 427–448. Psychology Press, 2007.

[81] J. Sullivan and S. Carlsson. Recognizing and tracking human action. In A. Heyden, G. S., M. Nielsen, and P. Johansen, editors, *ECCV (1)*, volume 2350 of *Lecture Notes in Computer Science*, pages 629–644. Springer, 2002.

[82] A M Treisman and G Gelade. A feature-integration theory of attention. *Cognit Psychol*, 12(1):97–136, January 1980.

[83] P. D. Turney and P. Pantel. From frequency to meaning: Vector space models of semantics. *J. Artif. Intell. Res. (JAIR)*, 37:141–188, 2010.

[84] K. Tzeras and S. Hartmann. Automatic indexing based on bayesian inference networks. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 22–35. ACM, 1993.

[85] N. Ueda and Z. Ghahramani. Bayesian model search for mixture models based on optimizing variational bounds. *Neural Networks*, 15(10):1223–1241, 2002.

[86] S. Veeramachaneni, D. Sona, and P. Avesani. Hierarchical Dirichlet model for document classification. In *Proceedings of ICML*, pages 928–935, 2005.

[87] J. Weston, S. Bengio, and N. Usunier. Large scale image annotation: learning to rank with joint word-image embeddings. *Machine Learning*, 81(1):21–35, 2010.

[88] J. Winn, C. M. Bishop, and T. Jaakkola. Variational message passing. *Journal of Machine Learning Research*, 6:661–694, 2005.

[89] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *Proceedings of ICCV*, volume 2, pages 1800–1807, Oct. 2005.

[90] T. Wong. Generalized Dirichlet distribution in bayesian analysis. *Applied Mathematics and Computation*, 97(2-3):165 – 181, 1998.