Concurrent and Continual Knowledge Transfer Using Latent Variable Models

Dissertation Proposal

Ayan Acharya

Supervising Professor:

Dr. Joydeep Ghosh

Co-supervising Professor:

Dr. Raymond J. Mooney

Committee:

Dr. Sanjay Shakkotai
Dr. Sujay Sanghavi
Dr. Suju Rajan

Department of Electrical and Computer Engineering
University of Texas at Austin

November 2013

Abstract

In several applications, scarcity of labeled data is a challenging problem that hinders the predictive capabilities of machine learning algorithms. Additionally, the distribution of the data changes over time rendering models trained with older data less capable of discovering useful structure from the newly available data.

Transfer learning is a convenient framework to overcome such problems where the learning of a model specific to a domain can benefit the learning of other models in other domains through either simultaneous training of domains or sequential transfer of knowledge from one domain to the others.

In this proposal, such learning methodologies are implemented using latent variable models. In all the approaches related to simultaneous or concurrent learning, a low dimensional space is maintained that is shared across multiple domains. For sequential or continual knowledge transfer, parameters of the model trained with data from an older domain are carefully adapted to fit the new distributions. Applications of such frameworks in supervised learning problems like text classification and object recognition from images have shown promising results so far. Concurrent knowledge transfer has also been integrated with active learning to gain additional benefits in domains where labeled data is hard to come by. In future, the following applications will be further explored: i) concurrent knowledge transfer with explicit feedback from human annotators, ii) large scale concurrent knowledge transfer, iii) document classification from multiple corpora, and iv) adaptation of unsupervised document categorization *via* continual knowledge transfer.

Contents

1 Introduction **4**

2 Background and Related Work **7**
2.1 Transfer and Multitask Learning . . . 7
2.2 Active Learning *via* Expected Error Reduction . . . 7
2.3 Active Knowledge Transfer . . . 8
2.4 Learning with Annotators' Rationale . . . 8
2.5 Online Support Vector Machines . . . 9
2.6 Non-negative Matrix Factorization (NMF) . . . 9
2.7 Statistical Topic Models . . . 10
2.8 Incremental EM Algorithm . . . 11
2.9 Variational Inference in Topic Models . . . 12
2.9.1 Batch Variational Inference in Topic Models . . . 12
2.9.2 Online Variational Inference in Topic Models . . . 13
2.10 Dynamic Topic Models (DTM) . . . 14
2.11 A Few Random Processes . . . 15
2.12 Hierarchical Dirichlet Process (HDP) . . . 17
2.12.1 Traditional Stick Breaking Construction of HDP Topic Model . . . 17
2.12.2 Modified Stick Breaking Construction of HDP Topic Model . . . 18
2.13 Generalized Negative Binomial Process (GNBP) . . . 18

3 Completed Work **20**
3.1 Multitask Learning Using Both Shared Latent and Supervised Topics . . . 20
3.1.1 Doubly Supervised Latent Dirichlet Allocation . . . 21
3.1.2 Non-parametric Doubly Supervised Latent Dirichlet Allocation . . . 22
3.2 Active Multitask Learning Using Both Shared Latent and Supervised Topics . . . 22
3.3 Transfer Learning using Probabilistic Combination of Classification and Clustering Ensembles 23

4 Proposed Work **26**
4.1 Active Multitask Learning using Annotators' Rationale . . . 26
4.2 Knowledge Transfer across Corpora of Different Languages for Category Classification . . . 28
4.3 Large Scale Multitask Learning using Topic Models . . . 29
4.4 Non-parametric Dynamic Topic Models . . . 31

5 Datasets for Empirical Evaluation **33**
5.1 aYahoo Data . . . 33
5.2 ACM Conference Data . . . 33
5.3 Science Data . . . 33

Bibliography **35**

A Publications **40**

B Schedule and Vita **42**

Attached Papers **43**

List of Figures

2.1 Schematic of Topic Models 10
2.2 LDA 10
2.3 LLDA 10
2.4 MedLDA 10
2.5 Smoothened LDA 14

3.1 DSLDA 21
3.2 NP-DSLDA 21
3.3 Combining Classifiers and Clusterers. 24

4.1 System Overview 29
4.2 NP-SLDAMC 30
4.3 First Scheme of Decomposition 30
4.4 Second Scheme of Decomposition 31

Chapter 1

Introduction

In several applications, scarcity of labeled data is a challenging problem that hinders the predicative capabilities of machine learning algorithms. Additionally, the distribution of the data changes over time rendering models trained with older data less capable of discovering useful structure from the newly available data. This proposal addresses the problems mentioned above in both text and vision domains and how to alleviate them with the help of *transfer learning* – a mechanism that exploits labeled information in one domain and uses it in others. Transfer learning allows the learning of models specific to a given domain to benefit from the learning of other models from other domains through either simultaneous (Caruana, 1997) or sequential (Bollacker and Ghosh, 2000) training. In this proposal, both these such learning methodologies are implemented using mixed-membership latent variable models.

For simultaneous or concurrent transfer learning, the proposed latent variable models have a common theme. All of them use some low dimensional representation of the data that is shared across different domains. The key idea is to project data from different domains onto a common low dimensional space so that the mapping from the original feature space to the low dimensional space is learnt better given data from multiple domains. The mapping from this low dimensional space to the target variables can then be learnt much more efficiently given only a few labeled information associated with the target variables. To further clarify the utility of a shared low dimensional space, we can consider a "baseline" scheme where data from different domains are projected onto disjoint low dimensional subspaces and the mapping to the target variables are learnt in an isolated manner. In such a scheme, if some domain has less labeled information, the mapping from the original feature space to the low dimensional space is not learnt well, which might result in poor predictive performance. In the literature of transfer learning, concurrent learning is popularly known as "multi-task learning" and isolated learning is sometimes referred to as "single-task learning".

Along the direction of concurrent learning using latent variable models, the following pieces of work are already completed:

1. Multitask learning using both shared latent and supervised topics.

2. Active multitask learning using both shared latent and supervised topics.

Both of these works, summarized in Sections 3.1 and 3.2, use a mixed-membership model named Latent Dirichlet Allocation (Blei et al., 2003) as a basic building block and the topics, which are distributions of words, are shared across data from different domains. This facilitates knowledge transfer across multiple domains. The target variables in these works are some class labels. For example, a class can be a particular conference and the data points pertaining to this class can be papers published in that conference. In the vision domain, the class might represent some object category and the data points pertaining to this class can be images belonging to that category. In the first work, partial supervision is allowed in the shared topic space and using both supervised and latent shared topics results in better knowledge transfer compared to baselines that do not use both types of shared topics. In the second work, concurrent knowledge transfer is integrated with active selection of class labels and supervised topics. Such integration is shown to be

very useful in domains where very few labeled information is available. Additionally, the models proposed query over both class labels and supervised topics which has not been explored in the context of concurrent knowledge before.

Based on the success of the above two works, three more projects are planned:

3. Active multitask learning using annotators' rationale.

4. Knowledge transfer across corpora of different languages for category classification.

5. Large scale multitask learning using topic models.

In the first problem, concurrent knowledge transfer is integrated with active learning where the annotators in the active learning process are required to provide some additional information regarding their choice of answers. For example, the annotator can highlight parts of text that he/she found relevant for categorizing a document into one of the classes or supervised topics. In the second problem, documents from multiple corpora are considered where knowledge transfer is achieved *via* a shared latent topic space. The problem derives its inspiration from industry where adequate amount of labeled information is available for webpages written in English but only a few labeled information is available for other foreign languages like Italian, French, Spanish and German. Transferring knowledge using a computationally efficient framework from the rich label set in English to other foreign languages seems to be a promising research direction. The third problem is concerned with discovering structure from large scale unsupervised data. In industry, monitoring behaviors of millions of users on different domains of the webpages and thereby improving the personalized recommendation can bring about massive revenue. Typically, each user has access to a set of documents which can be mapped onto a shared latent topic space. The users can be grouped according to the documents' assignment to the shared topic space and the topics can further be refined with the information from the user groups. Such shared representation and dimensionality reduction of both users and features are amenable for better personalized recommendation. Further details about these problems are discussed in Sections 4.1, 4.2 and 4.3 respectively.

For sequential or continual knowledge transfer, the common underlying theme is the training of the parameters of the associated latent variable models with data from an older domain and careful modification of these parameters for fitting the new distribution from the new domain(s). Along this direction the following two works are listed, of which the first one is completed and the second one is proposed.

6. Transfer learning using probabilistic combination of classification and clustering ensembles.

7. Non-parametric dynamic topic models.

A brief summary of the first work is provided in Section 3.3 and further details about the second project is illustrated in detail in Section 4.4. In the first work, it is assumed that some classifiers, built from the old domain, are employed to classify data from some new domain where the distribution of the data is different from the old domain. Therefore, the classifiers are expected to give less accurate predictions. The mixed-membership model proposed uses the classifier predictions and the clustering of the new data obtained by employing some unsupervised clustering algorithm and unifies these two information sources carefully. The challenge in this work is to balance the prediction from the supervised models and the unsupervised information. In the second work, a non-parametric dynamic topic model is proposed. In practice, the nature of the data changes over time and there might be newer topics emerging with time or old topics dying out (Ahmed and Xing, 2008). The parameters of the topic models should evolve with time to account for such changes. This work will deal with such variability in data with time and incremental adaptation of the model parameters to fit the new data.

The rest of the thesis is organized as follows. The related literature and the background works are presented in Section 2. The datasets that are used for evaluation in the completed works or are planned to be used in the proposed works are explained in Section 5.

Note on Notation: Vectors and matrices are denoted by bold-faced lowercase and capital letters, respectively. Scalar variables are written in italic font, and sets are denoted by calligraphic uppercase

letters. Pois(), Gamma(), Dir(), Beta() and multinomial() stand for Possion, Gamma, Dirichlet, Beta and multinomial distribution respectively. An indicator variable is defined as: $\mathbb{I}_{\mathcal{S},n} = 1$ if $n \in \mathcal{S}$ and $\mathbb{I}_{\mathcal{S},n} = 0$ if $n \notin \mathcal{S}$.

Chapter 2

Background and Related Work

In this chapter, the related works as well as the background materials are presented. Firstly, related literature on transfer learning and active learning is summarized. Afterwards, the relevant algorithms and models used in this proposal are illustrated. For an easy perusal, whenever convenient, different sections point to the chapters of completed and proposed works in which the concerned algorithms or models are used.

2.1 Transfer and Multitask Learning

Transfer learning allows the learning of some tasks to benefit the learning of others through either simultaneous (Caruana, 1997) or sequential (Bollacker and Ghosh, 2000) training. In multitask learning (MTL (Caruana, 1997)), a single model is simultaneously trained to perform multiple related tasks. MTL has emerged as a very promising research direction for various applications including biomedical informatics (Bickel et al., 2008), marketing (Evgeniou et al., 2007), natural language processing (Ando, 2006), and computer vision (Torralba et al., 2007). Many different MTL approaches have been proposed over the past 15 years (*e.g.*, see (Weinberger et al., 2009; Pan and Yang, 2010; Passos et al., 2012) and references therein). These include different learning methods, such as empirical risk minimization using group-sparse regularizers (Kim and Xing, 2010; Jenatton et al., 2011), hierarchical Bayesian models (Zhang et al., 2008; Low et al., 2011) and hidden conditional random fields (Quattoni et al., 2007). Evgeniou et al. (2005) proposed the regularized MTL which constrained the models of all tasks to be close to each other. The task relatedness in MTL has also been modeled by constraining multiple tasks to share a common underlying structure (Ben-David and Schuller, 2003; Argyriou et al., 2007; Caruana, 1997). Ando and Zhang (2005) proposed a structural learning formulation, which assumed multiple predictors for different tasks shared a common structure on the underlying predictor space.

In all of the MTL formulations mentioned above, the basic assumption is that all tasks are related. In practical applications, these might not be the case and the tasks might exhibit a more sophisticated group structure. Such structure is handled using clustered multi-task learning (CMTL). In (Bakker and Heskes, 2003) CMTL is implemented by considering a mixture of Gaussians instead of single Gaussian priors. Xue *et al.* (Xue et al., 2007) introduced the Dirichlet process prior that automatically identifies subgroups of related tasks. In (Jacob et al., 2008), a clustered MTL framework was proposed that simultaneously identified clusters and performed multi-task inference.

2.2 Active Learning *via* Expected Error Reduction

Of the several measures for selecting labels in active learning algorithms, a decision-theoretic approach called Expected Error Reduction (Roy and McCallum, 2001) has been used quite extensively in practice (Kovashka et al., 2011; Settles, 2009). This approach aims to measure how much the generalization error of a model is likely to be reduced based on some labeled information y of an instance $\boldsymbol{x}$ taken from the unlabeled pool

$\mathcal{U}$. The idea is to estimate the expected future error of a model trained using $\mathcal{L} \cup \langle \boldsymbol{x}, y \rangle$ on the remaining unlabeled instances in $\mathcal{U}$, and query the instance with minimal expected future error. Here $\mathcal{L}$ denotes the labeled pool of data. One approach is to minimize the expected 0/1 loss:

$$\boldsymbol{x}^*_{0/1} = \underset{\boldsymbol{x}}{\operatorname{argmax}} \sum_n P_{\boldsymbol{\kappa}}(y_n|\boldsymbol{x}) \left(\sum_{u=1}^{U} 1 - P_{\boldsymbol{\kappa}^{+\langle \boldsymbol{x}, y_n \rangle}} \left(\hat{y}, \boldsymbol{x}^{(u)} \right) \right). \tag{2.1}$$

where $\boldsymbol{\kappa}^{+\langle \boldsymbol{x}, y_n \rangle}$ refers to the new model after it has been re-trained with the training set $\mathcal{L} \cup \langle \boldsymbol{x}, y_n \rangle$. Note that we do not know the true label for each query instance, so we approximate using expectation over all possible labels under the current model. The objective is to reduce the expected number of incorrect predictions.

2.3 Active Knowledge Transfer

There has been some effort to integrate active and transfer learning in the same framework. In Jun and Ghosh (2008) the authors utilized a maximum likelihood classifier to learn parameters from the source domain and use these parameters to seed the EM algorithm that explains the unlabeled data in the target domain. The example which contributed to maximum expected KL divergence of the posterior distribution with the prior distribution was selected in the active step. In Rai et al. (2010), the source data was first used to train a classifier, the parameter of which was later updated in an online manner with new examples selected in the active step. The active selection criterion is based on uncertainty sampling (Settles, 2009). Similarly, in Chan and Ng (2007), a naïve Bayes classifier is first trained with examples from the source domain and then incrementally updated with the data from the target domain selected using uncertainty sampling. In Shi et al. (2008), the authors proposed to maintain a classifier trained from source domain(s) and the prediction of the classifier is trusted only when the likelihood of the data in the target domain is sufficiently high. In case of a lower likelihood value, domain experts are asked to label the example. The proposed method is independent of the active selection approach adopted. Harpale and Yang (2010) proposed active multitask learning for adaptive filtering (Robertson and Soboroff, 2002) where the underlying classifier is logistic regression with Dirichlet process priors. Any feedback provided in the active selection phase improves both the task-specific and the global performance *via* a measure called *utility gain* (Harpale and Yang, 2010). Saha et al. (2011) formulated an online active multitask learning framework where the information provided for one task is utilized for other tasks through a task correlation matrix. The updates are similar to perceptron updates. For active selection, they use a margin based sampling scheme which is a modified version of the sampling scheme used in Cesa-Bianchi et al. (2006).

The works presented in this thesis build on top of a topic model framework and use expected error reduction as active selection mechanism. Such active selection mechanism necessitates incremental update of the model parameters and hence the inference and estimation problems become challenging. This active selection mechanism is less immune to noisy observations compared to a simpler selection mechanism named uncertainty sampling (Settles, 2009).

2.4 Learning with Annotators' Rationale

In traditional supervised learning set-up, the annotators *only* provide the correct answer. However, the human annotators can better be employed to provide some rationale of choosing the answer. The authors in Zaidan et al. (2008); Donahue and Grauman (2011) showed that providing such additional information indeed improves the performances of the models. The basic framework entails an SVM learner. In case of a simple binary classification problem, a positive example is forced to maintain some gap with a negative example as well as the example derived from masking features of the positive example that an annotator

found relevant for labeling the example positive. Formally, the objective takes the following form:

$$\begin{aligned} & \frac{1}{2}||\boldsymbol{w}||^2 + C_1 \sum_n \xi_n + C_2 \sum_{n,m} \xi_{nm} \\ \text{s.t. } \forall n, \quad & y_n \langle \boldsymbol{w}, \boldsymbol{x}_n \rangle \geq (1 - \xi_n) \\ \forall n, \quad & \xi_n \geq 0 \\ \forall (n,m), \quad & y_n \langle \boldsymbol{w}, (\boldsymbol{x}_n - \boldsymbol{v}_{nm}) \rangle \geq \mu(1 - \xi_{nm}) \\ \forall (n,m), \quad & \xi_{nm} \geq 0 \end{aligned} \tag{2.2}$$

where, $\boldsymbol{x}_n$ is the n^{th} example, $y_n \in \{-1, +1\}$ is the label of the n^{th} example and $\boldsymbol{v}_{nm}$ is the example obtained from the positive example $\boldsymbol{x}_n$ by removing features found relevant for the example being positive by the m^{th} annotator. ξ_n and ξ_{nm} are the slack variables corresponding to the two different margin constraints, one of which is scaled by the parameter μ, thereby allowing more flexibility in the modeling. C_1 and C_2 determine how the margin violations would be penalized w.r.t the regularization term. Such learning mechanism is part of the work proposed in Section 4.1.

2.5 Online Support Vector Machines

The online SVM proposed by Bordes et al. (2007, 2005) has three distinct modules that work in unison to provide a scalable learning mechanism. These modules are named as "ProcessNew", "ProcessOld" and "Optimize". All of these modules use a common operation called "SMOStep" and the only memory footprint is due to the support vectors and the associated gradient information. The module "ProcessNew" operates on a pattern that is not a support pattern. In such an update, one of the classes is chosen to be the label of the support pattern and the other class is chosen in such a way that it defines feasible direction with the highest gradient. It then performs an SMO step with the example and the selected classes. The module "ProcessOld" randomly picks a support pattern and chooses two classes that define the feasible direction with the highest gradient for that support pattern. "Optimize" resembles "ProcessOld" but picks two classes among those that correspond to existing support vectors. Such online learning of SVM will be useful for incremental update of the parameters of the model proposed here in active selection step.

2.6 Non-negative Matrix Factorization (NMF)

Matrix factorization is a common dimensionality reduction approach, representing the original data using a lower dimensional latent space. A standard MF approach is to find two lower dimensional matrices such that when multiplied together approximately produce the original matrix. In such decomposition, the factors can have both positive and negative elements. However, in many applications, negative values are difficult to interpret. Positive matrix factorization (Paatero and Tapper, 1994), non-negative matrix factorization (Lee and Seung, 2000), and non-negative independent component analysis (Plumbley and Oja, 2004) are techniques that perform factorization in positively constrained components. Although these methods are fast and stable under relatively mild assumptions, they lack clear probabilistic generative semantics. Bayesian formulations of similar ideas have also been proposed (Miskin, 2000; Højen-Sørensen et al., 2002) in which the positivity is imposed using rectified Gaussian distribution (Socci et al., 1998), exponential distribution or a mixture of the two (Hoffman et al., 2010b). In particular in Cemgil (2009), element-wise gamma priors are placed to enforce the non-negativity constraint. Hoffman et al. (2010b) used a Gamma process prior to extend such model to accommodate infinite number of latent factors. The concepts of NMF will be useful while motivating the problem in Section 4.3.

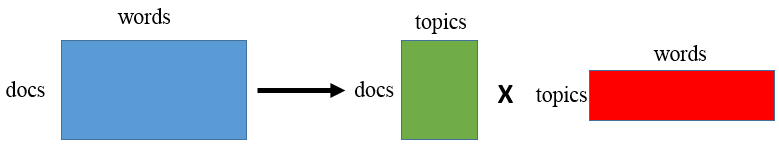

Figure 2.1: Schematic of Topic Models

2.7 Statistical Topic Models

LDA (Blei et al., 2003) treats documents as a mixture of topics, which in turn are defined by a distribution over a set of words. The words in a document are assumed to be sampled from multiple topics. In its original formulation, unsmoothened LDA can be viewed as a purely-unsupervised form of dimensionality reduction and clustering of documents in the topic space. On a high level, LDA can be thought of as performing a non-negative matrix factorization as shown in Fig. 2.1 (Buntine, 2002). The graphical model of LDA is shown in Fig 2.2. The generative process of LDA is described below:

- For the n^{th} document, sample a topic selection probability vector $\boldsymbol{\theta}_n \sim \text{Dir}(\boldsymbol{\alpha})$, where $\boldsymbol{\alpha}$ is the parameter of a Dirichlet distribution of dimension K, which is the total number of topics.
- For the m^{th} word in the n^{th} document, sample a topic $z_{nm} \sim \text{multinomial}(\boldsymbol{\theta}_n)$.
- Sample the word $w_{nm} \sim \text{multinomial}(\boldsymbol{\beta}_{z_{nm}})$, where $\boldsymbol{\beta}_k$ is a multinomial distribution over the vocabulary of words corresponding to the k^{th} topic.

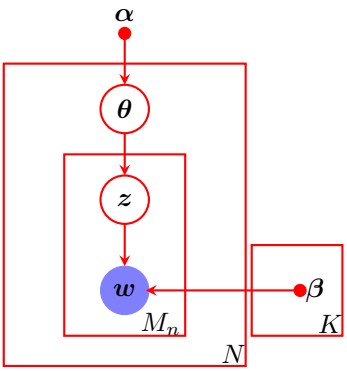

Figure 2.2: LDA

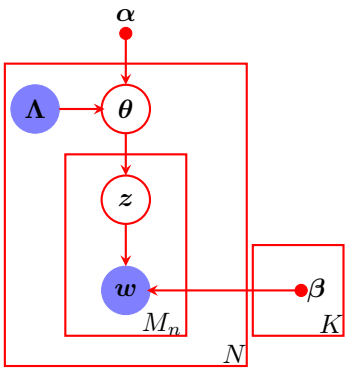

Figure 2.3: LLDA

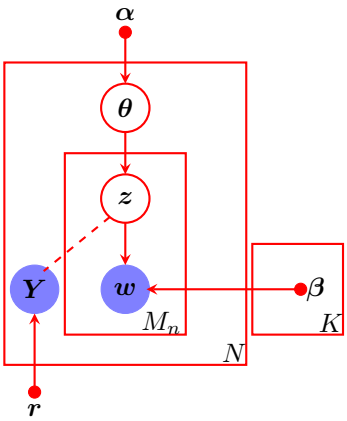

Figure 2.4: MedLDA

Several extensions of LDA have subsequently incorporated some sort of supervision. Some approaches provide supervision by labeling each document with its set of topics (Ramage et al., 2009; Rubin et al., 2011). In particular, in *Labeled LDA* (LLDA (Ramage et al., 2009)), the primary objective is to build a model of the words that indicate the presence of certain topic labels. For example, when a user explores a webpage based on certain tags, LLDA can be used to highlight interesting portions of the page or build a summary of the text from multiple webpages that share the same set of tags. The words in a given training document are assumed to be sampled *only* from the supervised topics, which the document has been labeled as covering. The graphical model of LLDA is shown in Fig. 2.3.

Some other researchers (Blei and Mcauliffe, 2007; Zhu et al., 2009; Chang and Blei, 2009) assume that supervision is provided for a single *response variable* to be predicted for a given document. The response

variable might be real-valued or categorical, and modeled by a normal, Poisson, Bernoulli, multinomial or other distribution (see Chang and Blei (2009) for details). Some examples of documents with response variables are essays with their grades, movie reviews with their numerical ratings, web pages with their number of hits over a certain period of time, and documents with category labels. In *Maximum Entropy Discriminative LDA* (MedLDA) (Zhu et al., 2009), the objective is to infer some low-dimensional (topic-based) representation of documents which is predictive of the response variable. Essentially, MedLDA solves two problems jointly – dimensionality reduction and max-margin classification using the features in the dimensionally-reduced space. In earlier versions of supervised topic models (Blei and Mcauliffe, 2007; Chang and Blei, 2009), categorical response variables were difficult to model since the resulting inference equations were complex. In particular, the use of Taylor's approximations breaks the guarantee that the likelihood lower bound increases after each update. Compared to earlier versions of supervised topic models (Blei and Mcauliffe, 2007; Chang and Blei, 2009), MedLDA has simpler update equations and produces superior experimental results. Therefore, in the frameworks presented in Sections 3.1 and 3.2, the max-margin principle adopted in MedLDA is preferred over other supervised topic models for modeling categorical response variables.

In MedLDA, the generative process of the words in the documents are same as the unsupervised LDA. However, the topic space representation of the documents is treated as features for an SVM learning framework. In particular, for the n^{th} document, we generate $Y_n = \arg\max_y \boldsymbol{r}_y^T \mathbb{E}(\bar{\mathbf{z}}_n)$ where Y_n is the class label associated with the n^{th} document, $\bar{\mathbf{z}}_n = \sum_{m=1}^{M_n} \mathbf{z}_{nm}/M_n$. Here, $\mathbf{z}_{nm}$ is an indicator vector of dimension K. $\boldsymbol{r}_y$ is a K-dimensional real vector corresponding to the y^{th} class, and it is assumed to have a prior distribution $\mathcal{N}(0, 1/C)$. M_n is the number of words in the n^{th} document. The maximization problem to generate Y_n (or the classification problem) is carried out using a max-margin principle – the exact formulation of which will be discussed later using variational approximation. Since MedLDA includes discriminative modeling of the class labels, it is not possible to draw a plate model. However, for the ease of understanding, in Fig. 2.4, we show a representative plate model with the discriminative part denoted by dotted lines.

2.8 Incremental EM Algorithm

The EM algorithm proposed by Dempster et al. (1977) can be viewed as a joint maximization problem over $q(.)$, the conditional distribution of the hidden variables $\boldsymbol{Z}$ given the model parameters $\boldsymbol{\kappa}$ and the observed variables $\boldsymbol{X}$. The relevant objective function is given as follows:

$$F(q, \boldsymbol{\kappa}) = \mathbb{E}_q[\log(p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\kappa}))] + H(q), \tag{2.3}$$

where $H(q)$ is the entropy of the distribution $q(.)$. Often, $q(.)$ is restricted to a family of distributions $\mathcal{Q}$. It can be shown that if $\boldsymbol{\kappa}^*$ is the maximizer of the above objective F then it also maximizes the likelihood of the observed data. Therefore, another representation of the t^{th} step of the EM algorithm is as follows:

- **E step:** $q^{(t)} = \underset{q}{\text{argmax}}\ F(q, \boldsymbol{\kappa}^{(t-1)})$.
- **M step:** $\boldsymbol{\kappa}^{(t)} = \underset{\boldsymbol{\kappa}}{\text{argmax}}\ F(q^{(t)}, \boldsymbol{\kappa})$.

In most of the models used in practice, the joint distribution is assumed to factorize over the instances implying that $p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\kappa}) = \prod_{n=1}^{N} p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\kappa})$. One can further restrict the family of distributions $\mathcal{Q}$ to maximize over in Eq. (2.3) to the factorized form: $q(\boldsymbol{Z}) = \prod_{n=1}^{N} q(\boldsymbol{z}_n|\boldsymbol{x}_n) = \prod_{n=1}^{N} q_n$.

An incremental variant of the EM algorithm that exploits such separability structure in both $p(.)$ and $q(.)$ was first proposed by Neal and Hinton (1999). Under such structure, the objective function in Eq.

(2.3) decomposes over the observations $F(q, \boldsymbol{\theta}) = \sum_{n=1}^{N} F_n(q_n, \boldsymbol{\kappa})$, and the following incremental algorithm can instead be used to maximize F:

- **E step:** Choose some observation n to be updated over, set $q_{n'}^{(t)} = q_{n'}^{(t-1)}$ for $n' \neq n$ (no update) and set $q_n^{(t)} = \underset{q_n}{\operatorname{argmax}}\ F_n(q_n, \boldsymbol{\kappa}^{(t-1)})$.
- **M step:** $\boldsymbol{\kappa}^{(t)} = \underset{\boldsymbol{\kappa}}{\operatorname{argmax}}\ F(q^{(t)}, \boldsymbol{\kappa})$.

Such an incremental view of the EM algorithm is useful for updating parameters in the proposed models in Section 3.2 for active query selection.

2.9 Variational Inference in Topic Models

The joint distribution over hidden and observed variables in unsupervised (unsmoothened) LDA can be written as follows:

$$p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\kappa}) = \prod_{n=1}^{N} p(\boldsymbol{\theta}_n|\boldsymbol{\alpha}) \prod_{m=1}^{M_n} p(z_{nm}|\boldsymbol{\theta}_n) p(w_{nm}|\boldsymbol{\beta}_{z_{nm}}) \tag{2.4}$$

The exact inference using EM is intractable due to the coupling between $\boldsymbol{\beta}$ and $\boldsymbol{Z}$ and hence variational EM is utilized. Here, the posterior distribution over the hidden variables is approximated by a completely factorized one as given below:

$$q(\boldsymbol{Z}|\boldsymbol{\kappa}^v) = \prod_{n=1}^{N} q(\boldsymbol{\theta}_n|\boldsymbol{\gamma}_n) \prod_{m=1}^{M_n} q(z_{nm}|\boldsymbol{\phi}_n) \tag{2.5}$$

Here, $\boldsymbol{\kappa}^v$ is the set of free variational parameters.

2.9.1 Batch Variational Inference in Topic Models

In batch variational inference for LDA, the following **E**vidence **L**ower **Bo**und (ELBO) is maximized w.r.t the model parameters $\boldsymbol{\kappa}$ and the variational parameters $\boldsymbol{\kappa}^v$:

$$\mathcal{L}(\boldsymbol{\kappa}^v, \boldsymbol{\kappa}) = \mathbb{E}_q[\log p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\kappa})] - \mathbb{E}_q[\log q(\boldsymbol{Z}|\boldsymbol{\kappa}^v)] \leq \log p(\boldsymbol{X}|\boldsymbol{\kappa}). \tag{2.6}$$

Using the factorization structure of both $p(.)$ and $q(.)$, one can see that the ELBO decomposes as follows:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\kappa}^v, \boldsymbol{\kappa}) &= \sum_{n=1}^{N} [\mathbb{E}_q[\log p(\boldsymbol{w}_n|\boldsymbol{z}_n, \boldsymbol{\beta})] + \mathbb{E}_q[\log p(\boldsymbol{\theta}_n|\boldsymbol{\alpha})] + \mathbb{E}_q[\log p(\boldsymbol{z}_n|\boldsymbol{\theta}_n)] \\ &\quad - \mathbb{E}_q[\log q(\boldsymbol{\theta}_n|\boldsymbol{\gamma}_n)] - \mathbb{E}_q[\log q(\boldsymbol{z}_n|\boldsymbol{\phi}_n)]] \\ &= \sum_{n=1}^{N} \ell(\boldsymbol{\kappa}_n^v, \boldsymbol{\kappa}). \end{aligned} \tag{2.7}$$

Here, $\ell(\boldsymbol{\kappa}_n^v, \boldsymbol{\kappa})$ denotes the contribution in the ELBO by the n^{th} document. $\mathcal{L}$ is optimized using coordinate ascent over each set of model and variational parameters. In the E-step, $\forall n$,$\ell(\boldsymbol{\kappa}_n^v, \boldsymbol{\kappa})$ is maximized w.r.t $\boldsymbol{\gamma}_n$ and $\boldsymbol{\phi}_n$. In the M step, the ELBO is maximized w.r.t the model parameters $\boldsymbol{\kappa}$. The algorithm, presented in 1, has constant memory requirements and empiricially converges faster than batch collapsed Gibbs sampling (Asuncion et al., 2009).

Algorithm 1 Batch Variational Bayes for LDA

Input: $\boldsymbol{X}$.
Output: $\boldsymbol{\kappa}$.

Initialize $\{\boldsymbol{\gamma}^n\}_{n=1}^N$, $\{\boldsymbol{\phi}_n\}_{n=1}^N$ randomly.
Until Convergence
E-Step
for $n = 1 : N$
  for $m = 1 : M_n$
    for $k = 1 : K$
      $\phi_{nmk} \propto \beta_{kw_{nm}} \exp(\Psi(\gamma_{nk}))$.
    end
    Normalize $\boldsymbol{\phi}_{nm}$.
    $\gamma_{nk} = \alpha_k + \sum_{m=1}^{M_n} \phi_{nmk}$.
  end
end
M-Step
Update $\boldsymbol{\alpha}$ using Newton-Raphson method.
$\beta_{kv} \propto \sum_{n=1}^{N} \sum_{m=1}^{M_n} \phi_{nmk} \mathbb{I}_{\{w_{nm=v}\}}$.
Normalize $\boldsymbol{\beta}_k$ $\forall k$.

With the use of the variational approximation mentioned above, the optimization problem in MedLDA (Zhu et al., 2009) takes the following form:

$$\min_{\boldsymbol{\kappa}^v, \boldsymbol{\kappa}, \{\xi_n\}} \frac{1}{2}||\boldsymbol{r}||^2 - \mathcal{L}(\boldsymbol{\kappa}^v, \boldsymbol{\kappa}) + C \sum_{n=1}^{N} \xi_n,$$
$$\text{s.t. } \forall n, y \neq Y_n, \mathbb{E}[\boldsymbol{r}^T \Delta f_n(y)] \geq (1 - \xi_n); \xi_n \geq 0, \tag{2.8}$$

where $\Delta f_n(y) = f(Y_n, \bar{\boldsymbol{z}}_n) - f(y, \bar{\boldsymbol{z}}_n)$ and $f(y, \bar{\boldsymbol{z}}_n)$ is a zero padded feature vector whose only non-zero components are the vectors in the position from $(y-1)K + 1$ to yK and equal $\bar{\boldsymbol{z}}_n$.

2.9.2 Online Variational Inference in Topic Models

Online inference in topic models follows from the theory of stochastic gradient descent applied to the ELBO in Eq. (2.7). (Hoffman et al., 2010a; Wang et al., 2011). The model considered for online inference is the smoothened LDA one shown in Fig. 2.5 where a Dirichlet prior with parameter $\boldsymbol{\eta}$ is placed over the topics. The following factorized approximation is used for smoothened LDA:

$$q(\boldsymbol{Z}|\boldsymbol{\kappa}^v) = \prod_{n=1}^{N} q(\boldsymbol{\theta}_n|\boldsymbol{\gamma}_n) \prod_{m=1}^{M_n} q(z_{nm}|\boldsymbol{\phi}_n) \prod_{k=1}^{K} q(\boldsymbol{\beta}_k|\boldsymbol{\lambda}_k). \tag{2.9}$$

Here, $q(\boldsymbol{\beta}_k|\boldsymbol{\lambda}_k)$ is a Dirichlet distribution of dimension V and with parameter $\boldsymbol{\lambda}_k$. The algorithm is presented in 2. The parameter $\rho_n = (\tau_0 + n)^{-\tau}$ weighs the contribution from the current document. $\tau \in (0.5, 1]$ is the learning rate used in stochastic gradient descent type of algorithm. $\tau_0 \geq 0$ slows down the learning rate. When infinite number of documents is considered, the global variational parameters $\{\boldsymbol{\lambda}_k\}_{k=1}^K$ are learnt by considering one document at a time and assuming that this document is a representative sample of the entire corpus with N documents and is replicated N times. In ideal situation, N should be infinity. However, for practical purpose, N is considered some large number.

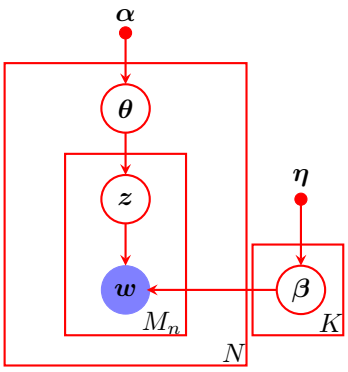

Figure 2.5: Smoothened LDA

2.10 Dynamic Topic Models (DTM)

For many collections of interest such as scholarly journals, email, news articles, and search query, the implicit assumption of exchangeable documents in topic models is inappropriate as the content evolves over time. For example, the Science article "The Brain of Professor Laborde" may be on the same scientific path as the article "Reshaping the Cortical MotorMap by Unmasking Latent Intracortical Connections" but the study of neuroscience looked much different in 1903 than it did in 1991 (Blei and Lafferty, 2006). The themes in a document collection evolve over time, and it is of interest to explicitly model the dynamics of the underlying topics.

Traditional time series modeling has only focused on continuous data. However, topic models only explain categorical data. Blei and Lafferty (2006) extended the vanilla LDA model to account for time evolving categorical data by using state space models on the natural parameters of the topic multinomials and also on the natural parameters of the logistic normal distributions used for modeling document specific topic selection. The generative process of this model is described below:

- $\forall k \in \{1, 2, \cdots, K\}$, sample $\boldsymbol{\beta}_{tk} \sim \mathcal{N}(\boldsymbol{\beta}_{(t-1)k}, \sigma^2 \boldsymbol{I})$ where $\boldsymbol{\beta}_{tk}$ is the natural parameter of the multinomial distribution over the k^{th} topic in the t^{th} time instant.
- Sample $\boldsymbol{\alpha}_t \sim \mathcal{N}(\boldsymbol{\alpha}_{(t-1)}, \delta^2 \boldsymbol{I})$ where $\boldsymbol{\alpha}_t$ is the natural parameter of the logistic normal distribution that is used for topic selection in the t^{th} time instant.
- For the n^{th} document in the t^{th} time step, sample a real vector $\boldsymbol{\theta}_{tn} \sim \mathcal{N}(\boldsymbol{\alpha}_t, a^2 \boldsymbol{I})$.
- For the m^{th} word in the n^{th} document in the t^{th} time step, sample a topic $z_{nmt} \sim \text{multinomial}(f(\boldsymbol{\theta}_{tn}))$. Here $f(.)$ is the softmax function.
- Sample the word $w_{nmt} \sim \text{multinomial}(f(\boldsymbol{\beta}_{z_{nmt}}))$, where $\boldsymbol{\beta}_k$ is the natural parameter of the multinomial distribution over the vocabulary of words corresponding to the k^{th} topic.

Inference in each time stamp in dynamic topic model is carried out using variational inference. For modeling the time dependence, either kalman filtering update equations (Kalman, 1960) or wavelet regression (Wasserman, 2005) is used. For convenience, this model will be denoted by discrete dynamic topic model (dDTM).

Several extensions of dDTM have been made so far. While the dDTM is a powerful model, the choice of time discretization affects the memory requirements and computational complexity of posterior inference. This largely determines the resolution at which to fit the model. Therefore, Wang et al. (2008) proposed a continuous time extension of the dynamic topic models (cDTM). The cDTM model, in general, introduces

Algorithm 2 Online Variational Bayes for Smoothened LDA

Input: $\boldsymbol{X}$.
Output: $\boldsymbol{\lambda}$.

Define $\rho_n = (\tau_0 + n)^{-\tau}$.
Initialize $\boldsymbol{\lambda}$ randomly.
for $n = 1 : \infty$
  E-Step
  Until Convergence
    for $m = 1 : M_n$
      for $k = 1 : K$

$$\phi_{nmk} \propto \exp\left(\Psi(\gamma_{nk}) + \mathbb{I}_{\{w_{nm}=v\}}\left[\Psi(\lambda_{kv}) - \Psi(\sum_{v=1}^{V}\lambda_{kv})\right]\right).$$

      end
      Normalize $\boldsymbol{\phi}_{nm}$.

$$\gamma_{nk} = \alpha_k + \sum_{m=1}^{M_n}\phi_{nmk}.$$

    end
  M-Step

$$\tilde{\lambda}_{kv} \propto \eta + N\sum_{m=1}^{M_n}\phi_{nmk}\mathbb{I}_{\{w_{nm=v}\}}\ .$$

  $\boldsymbol{\lambda} = (1-\rho_n)\boldsymbol{\lambda} + \rho_n\tilde{\boldsymbol{\lambda}}$.
  Normalize $\boldsymbol{\lambda}_k\ \forall k$.
end

many more latent variables than the dDTM. However, this seemingly more complicated model is simpler and more efficient to fit, as this formulation can take advantage of the natural sparsity of text, the fact that not all vocabulary words are used at each measured time step (Wang et al., 2008).

Ahmed and Xing (2008) introduced a non-parametric dDTM based on recurrent chinese restaurant process. This model addresses two shortcomings of the dDTM. The number of data points in each discrete time stamp can vary, for instance, data items may arrive or leave, or move in space (Wang et al., 2007) and this should be efficiently handled. The number of clusters may also vary over time and the model should be able to adjust its capacity accordingly. However, the proposed model used the posterior counts at time $(t-1)$ as the prior for topic proportion at time t, which does not follow the generative assumption of dDTM and is a heuristic adjustment. The inference for the topic mulltinomials does not contain contributions from time $(t+1)$, which is also problematic.

2.11 A Few Random Processes

Before explaining Hierarchical Dirichlet Process (HDP) and Generalized Negative Binomial Process (GNMP), which form the basis of some of the completed and proposed works, few definitions need to be introduced.

Definition – SumLog Distribution: m is defined to have a SumLog distribution with parameters (l, p) when $m = \sum_{t=1}^{l} u_t, u_t \sim \text{Log}(p)$. $u \sim \text{Log}(p)$is the logarithmic distribution with PMF $f_U(k) = -p^k/(k\ln(1-p))$. For conciseness, this will be written as $m \sim \sum Log(p)$.

Definition – Negative Binomial Distribution (NB): A negative Binomial distribution is obtained by placing a gamma distribution prior with shape r and scale $p/(1-p)$ on λ, the parameter of a Poisson distribution. Formally, $m \sim \text{Pois}(\lambda)$, $\lambda \sim \text{Gamma}(r, p/(1-p))$ By marginalizing out λ, a negative binomial distribution $m \sim \text{NB}(r, p)$ is obtained, with PMF:

$$f_M(m) = \frac{\Gamma(r+m)}{m!\Gamma(r)}(1-p)^r p^m, \tag{2.10}$$

where r is the nonnegative dispersion parameter and p is the probability parameter. For such construction, the NB distribution is also known as the gamma-Poisson mixture distribution (Greenwood and Yule, 1920).

Definition – Poisson Process: A Poisson process is defined as $X \sim \text{PP}(G_0)$ on the product space $\mathbb{Z}_+ \times \Omega$, with a finite continuous base measure G_0 over Ω, such that $X(A) \sim \text{Pois}(G_0(A))$ for each subset $\mathcal{A} \subset \Omega$.

Definition – Gamma Process (Wolpert et al., 2011): A Gamma process is defined as $X \sim \text{GaP}(G_0)$ on the product space $\mathbb{R}_+ \times \Omega$, with a finite continuous base measure G_0 over Ω, such that $X(A) \sim \text{Gamma}(c, G_0(\mathcal{A}), 1/c)$ for each subset $\mathcal{A} \subset \Omega$.

Definition – Dirichlet Process (Antoniak, 1974): Denote $\tilde{G} = G/G(\Omega)$, where $G \sim \text{GaP}(c, G_0)$, then for any measurable disjoint partition $\mathcal{A}_1, \cdots, \mathcal{A}_Q$ of Ω, we have

$$[\tilde{G}(\mathcal{A}_1), \cdots, \tilde{G}(\mathcal{A}_Q)] \sim \text{Dir}\big(\gamma_0 \tilde{G}_0(\mathcal{A}_1), \cdots, \tilde{G}_0(\mathcal{A}_Q)\big) \tag{2.11}$$

where, $\gamma_0 = G_0(\Omega)$ and $\tilde{G}_0 = G_0/\gamma_0$. With a space invariant concentration parameter, the normalized gamma process $\tilde{G} = G/G(\Omega)$ is a Dirichlet process with concentration parameter γ_0 and base probability measure $\tilde{G}_0$, expressed as $\tilde{G} \sim DP(\gamma_0, \tilde{G}_0)$.

Definition – Negative Binomial Process (NBP): To complete the Poisson process, a gamma process prior can be put over the base measure as follows:

$$\boldsymbol{X}_j \sim \text{PP}(G)\ \forall j \in \{1, 2, \cdots, J\},\ G \sim \text{GaP}(J(1-p)/p, G_0). \tag{2.12}$$

Marginalizing out G yields $X = \sum_j X_j \sim \text{NBP}(G_0, p)$, in which $X(A) \sim \text{NB}(G_0(\mathcal{A}), p)$ for each $\mathcal{A} \subset \Omega$.

Definition – Chinese Restaurant Process (CRP) (Pitman, 2006): In a Dirichlet process $\tilde{G} \sim DP(\gamma_0, \tilde{G}_0)$, it is assumed that $\boldsymbol{X}_i \sim \tilde{G}$, $\{\boldsymbol{X}_i\}$ are independent given $\tilde{G}$ and hence exchangeable. The predictive distribution of a new data point $\boldsymbol{X}_{(m+1)}$, conditioning on $\boldsymbol{X}_1, \cdots, \boldsymbol{X}_m$, with $\tilde{G}$ marginalized out, can be expressed as:

$$\boldsymbol{X}_{(m+1)} | \boldsymbol{X}_1, \cdots, \boldsymbol{X}_m \sim \mathbb{E}[\tilde{G} | \boldsymbol{X}_1, \cdots, \boldsymbol{X}_m] = \sum_{k=1}^{K} \frac{n_k}{m+\gamma_0}\delta_{\omega_k} + \frac{\gamma_0}{m+\gamma_0}\tilde{G}_0 \tag{2.13}$$

where $\{\omega_k\}_{k=1}^K$ are discrete atoms in Ω observed in $\boldsymbol{X}_1, \cdots, \boldsymbol{X}_m$ and $n_k = \sum_{i=1} \boldsymbol{X}_i(\omega_k)$ is the number of data points associated with ω_k. The stochastic process described in Eq. (2.13) is known as the Chinese restaurant process (Pitman, 2006).

Definition – Chinese Restaurant Table Distribution (CRT): Under the Chinese restaurant process metaphor, the number of data points m is assumed to be known whereas the number of distinct atoms K is treated as a random variable dependent on m and γ_0. Let $s(m, l)$ be the Stirling number of the first kind. Then it is shown in (Antoniak, 1974) that the random table count K has the following PMF:

$$\Pr(K = l | m, \gamma_0) = \frac{\Gamma(\gamma_0)}{\Gamma(m+\gamma_0)} |s(m, l)| \gamma_0^l,\ l = 0, 1, \cdots, m. \tag{2.14}$$

This distribution is referred to as the Chinese restaurant table (CRT) distribution and denoted by $l \sim \text{CRT}(m, \gamma_0)$, a CRT random variable.

Definition – Chinese Restaurant Table Process (CRTP): Define $L \sim \text{CRTP}(X, G_0)$ as a CRT process where for each $\mathcal{A} \subset \Omega$, $L(\mathcal{A}) = \sum_{\omega \in \mathcal{A}} L(\omega)$, $L(\omega) \sim \text{CRT}(X(\omega), G_0(\omega))$.

2.12 Hierarchical Dirichlet Process (HDP)

HDP has been a convenient tool for sharing clusters among multiple related groups. Its generative process is given below:

$$\begin{aligned} G_0|\gamma_0, H &\sim \text{DP}(\gamma_0, H) \\ G_j|\alpha_0, G_0 &\sim \text{DP}(\alpha_0, G_0)\ \forall j, \end{aligned} \tag{2.15}$$

where j indexes the group. There are multiple explanations available for Hierarchical Dirichlet Process, namely the Pólya urn model (Blackwell and Macqueen, 1973), Chinese restaurant process and the stick breaking process (Teh et al., 2006; Sethuraman, 1994). The last one is adopted here as this explanation is useful for developing variational inference for topic models which use HDP as prior. In what follows, the state-of-the art realization of HDP using stick breaking construction is presented first. Thereafter, a modification to this construction is illustrated that makes variational inference feasible and only this modified stick breaking representation will be used in Sections 3.1, 3.2 and 4.2.

2.12.1 Traditional Stick Breaking Construction of HDP Topic Model

An explicit representation of a draw from a DP was given by Sethuraman (1994), who showed that if $G \sim \text{DP}(\alpha_0; G_0)$, then with probability one: $G = \sum_{k=1}^{\infty} \beta_k \delta_{\phi_k}$ where the ϕ_k's are independent random variables distributed according to G_0 and δ_{ϕ_k} is an atom at ϕ_k. The "stick-breaking weights" β_k are random and depend on the parameter α_0. The representation above shows that draws from a DP are discrete with probability one. This discrete nature of the DP makes it suitable for the problem of placing priors on mixture components in mixture modeling where a mixture component can be associated with each atom in G.

To force G_0 to be discrete and yet have broad support, G_0 itself is drawn from a Dirichlet process $\text{DP}(\gamma_0, H)$. The atoms in ϕ_k are shared among the multiple DPs, yielding the desired sharing of atoms among groups (Teh et al., 2006). For HDP topic model, the lower level measures correspond to each document. The generative process for the first stage of HDP topic model goes as follows:

- $\forall k \in \{1, 2, \cdots, \infty\}$, generate $\beta'_k \sim \text{Beta}(1, \gamma_0)$. Set $\beta_k = \beta'_k \prod_{j=1}^{k-1}(1 - \beta'_j)$.
- Generate $\phi_k \sim H$, where H is some distribution.
- Create the base distribution G_0 following the summation: $G_0 = \sum_{k=1}^{\infty} \beta_k \delta_{\phi_k}$.

The generative process for the second stage is illustrated below:

- $\forall t \in \{1, 2, \cdots, \infty\}$, generate $\pi'_{nt} \sim \text{Beta}(\alpha_0 \beta_t, \alpha_0(1 - \sum_{l=1}^{t} \beta_l))$. Set $\pi_{nt} = \pi'_{nt} \prod_{l=1}^{t-1}(1 - \pi'_{nl})$.
- Create document specific distribution $G_n = \sum_{t=1}^{\infty} \pi_{nt} \delta_{\phi_t}$. Note that here π'_{nt}'s are generated in a way so that there is some sharing of atom weights across the DPs corresponding to different documents. This sharing of parameters is essential for maintaining the hierarchical structure assumed in the parametric LDA model.

Any non-parametric topic model that uses this particular view of HDP uses either gibbs sampling (Teh et al., 2006) or collapsed variational inference (Teh et al., 2007) for inference.

2.12.2 Modified Stick Breaking Construction of HDP Topic Model

Wang et al. (2011) proposed the following modification to the second stage of the stick breaking construction of HDP as shown below:

- $\forall t \in \{1, 2, \cdots, \infty\}$, generate $\psi_{nt} \sim G_0$.
- $\forall t \in \{1, 2, \cdots, \infty\}$, generate $\pi'_{nt} \sim \text{Beta}(1, \alpha_0)$. Set $\pi_{nt} = \pi'_{nt} \prod_{l=1}^{t-1}(1 - \pi'_{nl})$.
- Create data specific distribution: $G_n = \sum_{t=1}^{\infty} \pi_{nt} \delta_{\psi_{nt}}$.

Essentially, this sampling process avoids the complicated sharing of atom weights in lower level measures and suggests a new method which is amenable for variational inference without compromising the hierarchical structure. Interested reader can explore Wang et al. (2011); Wang and Carin (2012) for a more comprehensive discussion of this modification.

2.13 Generalized Negative Binomial Process (GNBP)

Although the number of points assigned to clusters are counts, mixture modeling is not typically considered as a count-modeling problem. Instead, clustering is often performed using the Dirichlet-multinomial framework, employing the Dirichlet process (Neal, 2000; Antoniak, 1974) as the prior distribution. Dirichlet process mixture models enjoy tractability as Dirichlet distribution is conjugate to multinomials. Despite its popularity, Dirichlet process is inflexible. A single concentration parameter controls the variability of the mass around the mean (Teh, 2010; Wolpert et al., 2011). Additionally, the inference of the concentration parameter is nontrivial and is usually solved with the data augmentation method proposed in Escobar and West (1995).

In order to construct more expressive mixture models with tractable inference algorithms, Zhou and Carin (2013) consider mixture modeling as a count-modeling problem. Directly modeling the counts assigned to clusters as NB random variables, Zhou and Carin (2013) perform joint count and mixture modeling *via* the NBP, using completely random measures (Kingman, 1967; Jordan, 2010) that are simple to construct and amenable for posterior computation. The following two lemmata are useful for describing the work of Zhou and Carin (2013).

Lemma 2.13.1 (Zhou and Carin (2013))**.** *The Poisson-logarithmic bivariate count distribution with PMF* $f_{M,L}(m, l|r, p) = \frac{|s(m,l)|r^l}{m!}(1-p)^r p^m$ *can be expressed as the product of a CRT and an NB distribution and also the product of a SumLog and a Poisson distributions as*

$$PoisLog(m, l; r, p) = CRT(l; m, r)NB(m; r, p) = SumLog(m; l, p)Pois(l; -r ln(1-p)). \tag{2.16}$$

Lemma 2.13.2 (Zhou and Carin (2013))**.** *Let* $m \sim NB(r, p)$, $r \sim Gamma(r_1, 1/c_1)$ *represent the gamma-NB distribution, denote* $p' = \frac{-ln(1-p)}{c_1 - ln(1-p)}$, *then* m *can also be generated from a compound distribution as* $m \sim \sum_{t=1}^{l} Log(p)$, $l \sim \sum_{t'=1}^{l'} Log(p')$, $l' \sim Pois(-r_1 ln(1-p'))$ *which is equivalent in distribution to* $m \sim \sum_{t=1}^{l} Log(p), l' \sim CRT(l, r_1), l \sim NB(r_1; p')$.

For joint count and mixture modeling of grouped data, Zhou and Carin (2013) share the NB dispersion while the probability parameters are group dependent. A gamma-NB process is constructed as follows:

$$\boldsymbol{X}_j \sim \text{NBP}(G, p_j), G \sim \text{GaP}(c, G_0). \tag{2.17}$$

The gamma-NB process can be augmented as a gamma-gamma-Poisson process as

$$\boldsymbol{X}_j \sim PP(\Theta_j), \Theta_j \sim \text{GaP}((1-p_j)/p_j, G), G \sim \text{GaP}(c, G_0) \tag{2.18}$$

This construction introduces gamma random measures Θ_j based on G, which are essential to construct group-level probability measures $\tilde{\Theta}_j$ to assign observations to mixture components. The gamma-NB process can also be augmented under the compound Poisson representation as

$$X_j \sim \sum_{t=1}^{L_j} \text{Log}(p_j), L_j \sim \text{PP}(-G\ln(1-p_j)), G \sim \text{GaP}(c, G_0) \tag{2.19}$$

which, using Lemma 2.13.1, is equivalent in distribution to

$$L_j \sim CRTP(\boldsymbol{X}_j, G), \boldsymbol{X}_j \sim \text{NBP}(G, p_j), G \sim \text{GaP}(c, GG_0) \tag{2.20}$$

Using Lemma 2.13.1 and 2.13.2, two equivalent augmentations can be proposed:

$$L \sim \sum_{t=1}^{L'} \text{Log}(p'), L' \sim \text{PP}(-G_0\ln(1-p_j)),, G \sim \text{GaP}(c, G_0) \tag{2.21}$$

$$L' \sim \text{CRTP}(L, G_0), L \sim \text{NBP}(G_0, p_0) \tag{2.22}$$

where $L = \sum_j L_j$. These augmentations allow to derive a sequence of closed-form update equations for inference. Such joint count and mixture modeling will be used as a basic building block for the model proposed in Section 4.4.

Chapter 3

Completed Work

This chapter presents the summaries of the three sets of completed work. The relevant papers are attached at the end of the proposal.

3.1 Multitask Learning Using Both Shared Latent and Supervised Topics

Humans can distinguish as many as 30,000 relevant object classes (Biederman, 1987). Training an isolated object detector for each of these different classes would require millions of training examples in aggregate. Computer vision researchers have proposed a more efficient learning mechanism in which object categories are learned via *shared* attributes, abstract descriptors of object properties such as "striped" or "has four legs" (Farhadi et al., 2009; Lampert et al., 2009; Kovashka et al., 2011). The attributes serve as an intermediate layer in a classifier cascade. The classifier in the first stage is trained to predict the attributes from the raw features and that in the second stage is trained to predict the categories from the attributes. During testing, only the raw features are observed and the attributes must be inferred. This approach is inspired by human perception and learning from high-level object descriptions. For example, from the phrase "eight-sided red traffic sign with white writing", humans can detect stop signs (Lampert et al., 2009). Similarly, from the description "large gray animals with long trunks", human can identify elephants. If the *shared* attributes transcend object class boundaries, such a classifier cascade is beneficial for *transfer learning* (Pan and Yang, 2010) where fewer labeled examples are available for some object categories compared to others (Lampert et al., 2009).

In Multitask learning (MTL), if the tasks are related, training one task should provide helpful "inductive bias" for learning the other tasks. To enable the reuse of training information across multiple related tasks, all tasks might utilize the same latent shared intermediate representation – for example, a shared hidden layer in a multi-layer perceptron (Caruana, 1997). In this case, the training examples for all tasks provide good estimates of the weights connecting the input layer to the hidden layer, and hence only a small number of examples per task is sufficient to achieve high accuracy. This approach is in contrast to "isolated" training of tasks where each task is learned independently using a separate classifier.

In the work of (Acharya et al., 2013b), the objective is to combine these two approaches to build an MTL framework that can use *both* attributes *and* class labels. The multiple tasks here correspond to different object categories (classes), and *both* observable attributes and latent properties are shared across the tasks. It is hereby emphasized that the proposed frameworks support general MTL; however, the datasets used happen to be multiclass, where each class is treated as a separate "task" (as typical in multi-class learning based on binary classifiers). But, in no way are the frameworks restricted to multiclass MTL. Since attribute-based learning has been shown to support effective transfer learning in computer vision, the tasks here naturally correspond to object classes.

MedLDA (Zhu et al., 2009) is used as a basic building block in the models proposed in this paper. These models are named Doubly Supervised Latent Dirichlet Allocation (DSLDA) and Non-parametric Doubly Supervised Latent Dirichlet Allocation (NP-DSLDA). Fig. 3.1 and 3.2 respectively illustrate the models. Note that these are not proper plate models following the argument given in case of MedLDA in Section 2.7.

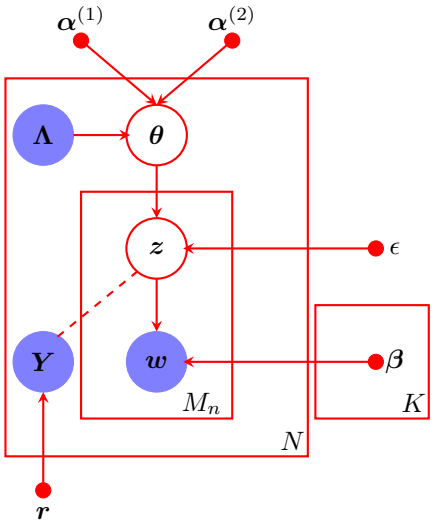

Figure 3.1: DSLDA

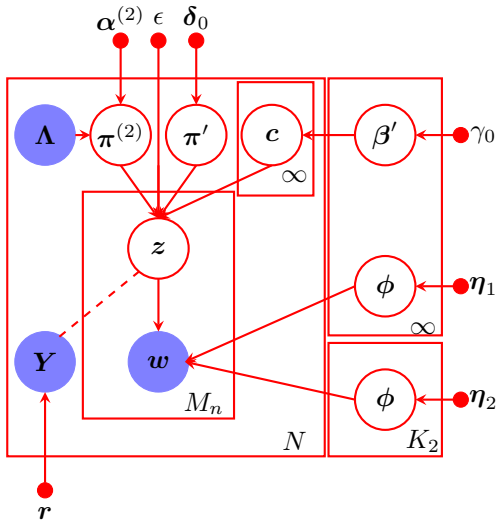

Figure 3.2: NP-DSLDA

Basically, two different sets of topics are maintained in each of these models. The first set corresponds to latent topics and the second set corresponds to supervised topics. All the tasks share the same set of latent topics and the same global set of supervised topics. On a high level, transfer happens when documents from all the tasks are projected onto this shared topic space. We explain the both of the models DSLDA and NP-DSLDA below.

3.1.1 Doubly Supervised Latent Dirichlet Allocation

• For the n^{th} document, sample a topic selection probability vector $\boldsymbol{\theta}_n \sim \text{Dir}(\boldsymbol{\alpha}_n)$, where $\boldsymbol{\alpha}_n = \boldsymbol{\Lambda}_n\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}$ is the parameter of a Dirichlet distribution of dimension K, which is the total number of topics. The topics are assumed to be of two types – latent and supervised, and there are K_1 latent topics and K_2 supervised topics. Therefore, $K = K_1 + K_2$. Latent topics are never observed, while supervised topics are observed in training but not in test data. Henceforth, in each vector or matrix with K components, it is assumed that the first K_1 components correspond to the latent topics and the next K_2 components to the supervised topics. $\boldsymbol{\Lambda}_n$ is a diagonal binary matrix of dimension $K \times K$. The k^{th} diagonal entry is unity if *either* $1 \leq k \leq K_1$ *or* $K_1 < k \leq K$ and the n^{th} document is tagged with the k^{th} topic. Also, $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2)$ where $\boldsymbol{\alpha}_1$ is a parameter of a Dirichlet distribution of dimension K_1 and $\boldsymbol{\alpha}_2$ is a parameter of a Dirichlet distribution of dimension K_2.

• For the m^{th} word in the n^{th} document, sample a topic $z_{nm} \sim \text{multinomial}(\boldsymbol{\theta}'_n)$, where $\boldsymbol{\theta}'_n = (1 - \epsilon)\{\boldsymbol{\theta}_{nk}\}_{k=1}^{k_1}\epsilon\{\Lambda_{n,kk}\boldsymbol{\theta}_{nk}\}_{k=1+k_1}^{K}$. This implies that the supervised topics are weighted by ϵ and the latent topics are weighted by $(1 - \epsilon)$. Sample the word $w_{nm} \sim \text{multinomial}(\boldsymbol{\beta}_{z_{nm}})$, where $\boldsymbol{\beta}_k$ is a multinomial distribution over the vocabulary of words corresponding to the k^{th} topic.

• For the n^{th} document, generate $Y_n = \arg\max_y \boldsymbol{r}_y^T \mathbb{E}(\bar{\mathbf{z}}_n)$ where Y_n is the class label associated with the n^{th} document, $\bar{\mathbf{z}}_n = \sum_{m=1}^{M_n} \mathbf{z}_{nm}/M_n$. Here, $\mathbf{z}_{nm}$ is an indicator vector of dimension K. $\boldsymbol{r}_y$ is a K-dimensional

real vector corresponding to the y^{th} class, and it is assumed to have a prior distribution $\mathcal{N}(0, 1/C)$. M_n is the number of words in the n^{th} document. The maximization problem to generate Y_n (or the classification problem) is carried out using a max-margin principle similar to MedLDA.

3.1.2 Non-parametric Doubly Supervised Latent Dirichlet Allocation

• Sample $\boldsymbol{\phi}_{k_1} \sim \text{Dir}(\boldsymbol{\eta}_1)\ \forall k_1 \in \{1, 2, \cdots, \infty\}$ and $\boldsymbol{\phi}_{k_2} \sim \text{Dir}(\boldsymbol{\eta}_2)\ \forall k_2 \in \{1, 2, \cdots, K_2\}$. $\boldsymbol{\eta}_1$, $\boldsymbol{\eta}_2$ are the parameters of Dirichlet distribution of dimension V.

• Sample $\beta'_{k_1} \sim \text{Beta}(1, \delta_0)\ \forall k_1 \in \{1, 2, \cdots, \infty\}$.

• For the n^{th} document, sample $\boldsymbol{\pi}_n^{(2)} \sim \text{Dir}(\boldsymbol{\Lambda}_n \boldsymbol{\alpha}_2)$. $\boldsymbol{\alpha}_2$ is the parameter of Dirichlet of dimension K_2. $\boldsymbol{\Lambda}_n$ is a diagonal binary matrix of dimension $K_2 \times K_2$. The k^{th} diagonal entry is unity if the n^{th} word is tagged with the k^{th} supervised topic.

• $\forall n, \forall t \in \{1, 2, \cdots, \infty\}$, sample $\pi'_{nt} \sim \text{Beta}(1, \alpha_0)$. Assume $\boldsymbol{\pi}_n^{(1)} = (\pi_{nt})_t$ where $\pi_{nt} = \pi'_{nt} \prod_{l<t}(1 - \pi'_{nl})$.

• $\forall n, \forall t$, sample $c_{nt} \sim \text{multinomial}(\boldsymbol{\beta})$ where $\beta_{k_1} = \beta'_{k_1} \prod_{l<k_1}(1 - \beta'_l)$. $\boldsymbol{\pi}_n^{(1)}$ represents the probability of selecting the sampled atoms in $\boldsymbol{c}_n$. Due to sampling with replacement, $\boldsymbol{c}_n$ can contain multiple atoms of the same index from the corpus level DP.

• For the m^{th} word in the n^{th} document, sample $z_{nm} \sim \text{multinomial}((1 - \epsilon)\boldsymbol{\pi}_n^{(1)}, \epsilon\boldsymbol{\pi}_n^{(2)})$. This implies that w.p. ϵ, a topic is selected from the set of supervised topics and w.p. $(1 - \epsilon)$, a topic is chosen from the set of (infinite number of) unsupervised topics. Note that by weighting the $\boldsymbol{\pi}$'s appropriately, the need for additional hidden "switching" variable is avoided.

• Sample w_{nm} from a multinomial given by the following equation:

$$\prod_{k_1=1}^{\infty} \prod_{v=1}^{V} \phi_{k_1 v}^{\mathbb{I}\{w_{nm}=v\}\mathbb{I}\{c_{nz_{nm}}=k_1 \in \{1,\cdots,\infty\}\}} \prod_{k_2=1}^{K_2} \prod_{v=1}^{V} \phi_{k_2 v}^{\mathbb{I}\{w_{nm}=v\}\mathbb{I}\{z_{nm}=k_2 \in \{1,\cdots,K_2\}\}}. \tag{3.1}$$

The results on aYahoo and Conference abstract data (described in Chapter 5) support the hypothesis that DSLDA's ability to incorporate both supervised and latent topics allow it to achieve better predictive performance compared to baselines that exploit only one, the other, or neither. Furthermore, NP-DSLDA is shown to be able to automate model-selection, performing nearly as well as DSLDA with optimally chosen parameters.

3.2 Active Multitask Learning Using Both Shared Latent and Supervised Topics

Another direction of improving the capacity of automated object detector is to employ intelligent label selection mechanism. Therefore, a group of computer vision researchers have formulated methods based on active learning for reducing the expense of human annotations where the automated system can request labels for the most informative examples (Qi et al., 2008; Jain and Kapoor, 2009; Joshi et al., 2009; Kovashka et al., 2011). In (Acharya et al., 2014), the objective is to combine two orthogonal approaches – learning from a shared topic space like the works in (Acharya et al., 2013b) and making active queries over both supervised topics and class labels. Two models are proposed here – Active Doubly Supervised LAtent Dirichlet Allocation (Act-DSLDA) and Active Non-parametric Doubly Supervised LAtent Dirichlet Allocation (Act-NPDSLDA).

Although the models build on the works in (Acharya et al., 2013b), the changes are not incremental. Firstly, one should note that in the test data, the supervised topics are not observed and one has to infer them

from either the parameters of the model or use some other auxiliary information. Since one of our objectives is to query over the supervised topics as well as the final category, we train a set of binary SVM classifiers that can predict the individual attributes from the features of the data. We denote the parameters of such classifiers by $\{\boldsymbol{r}_{2k}\}_{1\leq k\leq K_2}$. This is important to get an uncertainty measure over the supervised topics. To further clarify the issue, let us consider that only one supervised topic has to be labeled by the annotator for the n^{th} document from the set of supervised topics of size K_2. To select the most uncertain topic, one needs to compare the uncertainty of predicting the presence or absence of the individual topics. This uncertainty is different from that calculated from the conditional distribution calculated from the posterior over θ_n in the model DSLDA.

The inference and learning mechanism for Act-DSLDA in the batch mode is similar to the inference and learning in DSDLA except the fact that online SVM is used for max-margin learning. This is essential to maintain the support vectors and incrementally update them in the active selection step. From the modeling perspective, the difference between DSLDA (Acharya et al., 2013b) and Act-DSLDA lies in maintaining attribute classifiers and ignoring documents in the max-margin learning that do not have any class label.

The method of Expected Entropy Reduction requires one to take a data point from the unlabeled pool and one of its possible labels, update the model, and observe the generalized error on the unlabeled pool. This process is computationally expensive unless there is an efficient way to update the model incrementally. The incremental view of EM and the online SVM framework are appropriate for such update.

Consider that a completely unlabeled or partially labeled document, indexed by n', is to be included in the labeled pool with one of the $(K_2 + 1)$ labels (one for the class label and each different supervised topic), indexed by k'. In the E step, variational parameters corresponding to all other documents except for the n'th one is kept fixed and the variational parameters for only the n'th document are updated. In the M-step, we keep the priors $\{\boldsymbol{\alpha}^{(1)}, \boldsymbol{\alpha}^{(2)}\}$ over the topics and the SVM parameters $\boldsymbol{r}_2$ fixed as there is no easy way to update such parameters incrementally. From the empirical point of view, these parameters do not change much w.r.t. the variational parameters (or features in topic space representation) of a single document. However, the update of the parameters $\{\boldsymbol{\beta}, \boldsymbol{r}_1\}$ is easier. Updating $\boldsymbol{\beta}$ is accomplished by a simple update of the sufficient statistics. Updating $\boldsymbol{r}_1$ is done using the "ProcessNew" operation of online SVM followed by a few iterations of "ProcessOld". The selection of the document-label pair is guided by the measure given in Eq. (2.1). Note that since SVM uses hinge loss which, in turn, upper bounds the $0-1$ loss in classification, use of the measure from Eq. (2.1) for active query selection is justified. The NP-DSLDA model is also modified similarly to make Act-NPDSDLA work and the details can be found in the attached paper. Experimental results on both multi-class text and image datasets comparing to six different ablations of these models demonstrate the utility of integrating active and multitask learning in one framework that also unifies latent and supervised shared topics.

3.3 Transfer Learning using Probabilistic Combination of Classification and Clustering Ensembles

In many practical applications, the data changes over time in a very natural way. For example, users of eBay Inc. change the way they sell their items on the website by modifying the title of the items over time. The content of the images taken from satellites of different parts of earth's surface changes over seasons (popularly known as remotely sensed images). Since acquisition of labeled information for drifting data is often very time consuming and costly, a better design choice is to use prediction from the classifiers already trained with old data and adapt the predictions according to the unsupervised information available from the new unlabeled data. Unsupervised models can provide supplementary soft constraints to help classify new unlabeled data under the assumption that similar objects in the target set are more likely to share the same class label. (Acharya et al., 2013a) describe a Bayesian framework that takes as input class labels from existing classifiers (designed based on labeled data from the source domain), as well as cluster labels from a cluster ensemble operating solely on the target data to be classified, and yields a consensus labeling of the target data. The framework is illustrated in Fig. 3.3.

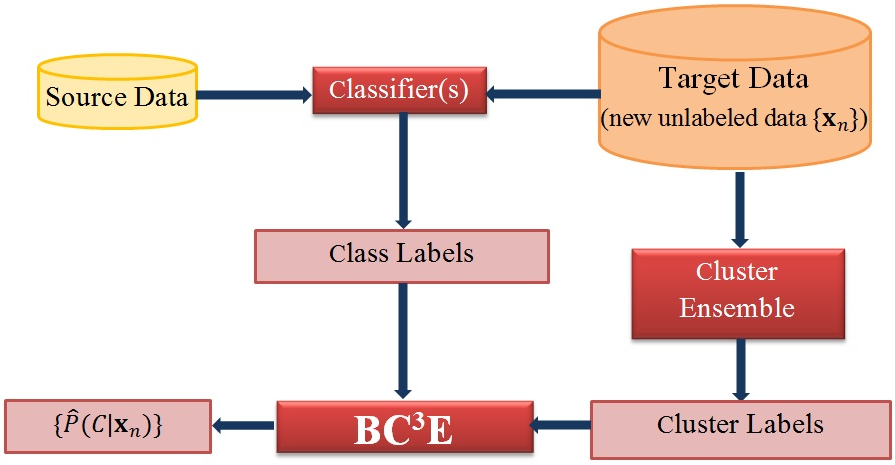

Figure 3.3: Combining Classifiers and Clusterers.

In practice, the observed class labels and the cluster labels carry different intrinsic weights. If the observations from the classifiers are assigned too much weight compared to those from clustering, there is little hope for the clustering to enhance classification. Similarly, if the observations from the clustering are given too much of importance, the classification performance might deteriorate. Ideally, the unsupervised information is only expected to enhance the classification accuracy. Therefore, a "safe" model is proposed that can intelligently utilize or reject the unsupervised information by learning a particular variance parameter from the clustering and classification results.

The proposed model, however, is much more general and can be applied for semi-supervised learning as well. As a particular application where the data changes over time, we use the item categorization problem of eBay Inc. The training data for this categorization problem consists of 83 million items sold over a three month period of time and the test set contains several millions of items sold a few days after the training period. More details about the dataset can be found in (Shen et al., Oct. 2012). eBay organizes items into a six-level category structure where there are 39 top level nodes called *meta categories* and 20K+ bottom level nodes called *leaf categories*. The dataset is generated when users provide the titles of items they intend to sell on eBay. Each title is limited to 50 characters, based on which the user gets recommendation of some *leaf categories* the item should belong to. Such categorization of the item helps a seller list an item in the correct branch of the product list, thereby allowing a buyer more easily search through a list of few million items sold via eBay every single day. A carefully designed k-Nearest Neighbor (k-NN) classifier (with the help of improved search engine algorithms) categorizes each of the items in less than 100 ms (Shen et al., Oct. 2012). However, due to the large number of categories (20K), items belonging to similar types of categories often get misclassified.

To avoid such confusion among conceptually similar categories, larger categories are formed by aggregating examples from categories that are relatively difficult to separate. Such aggregation is easy once the confusion matrix of the classification, obtained from a development dataset, is partitioned and strongly connected vertices (each vertex representing one of 20K *leaf categories*) are identified from the confusion graph, thereby forming a set of cliques which represent the large categories. Note that the large categories so discovered might not at all follow the internal hierarchy that is maintained. Next, clustering is performed with examples belonging to each of the large categories and the clustering results, along with the predictions from k-NN classification, are fed to BC^{3}E. The idea here is to first reduce the classification space and then use unsupervised information to refine the predictions from k-NN on a smaller number of categories. The number of *leaf categories* belonging to such large categories usually varies between 4-10.

However, the dataset is very dynamic and, typically over a span of three months, 20% of new words are added to the existing vocabulary. One can retrain the existing k-NN classifier every three months, but

the training process requires collecting new labeled data which is time consuming and expensive. One can additionally design classifiers to segregate examples belonging to each of the large categories. However, such approach might not improve much upon the performance of the initial k-NN classifier if the data changes so frequently, which necessitates a system that can adaptively predict newer examples without retraining the existing classifier or employing another set of classification algorithms. BC^{3}E is useful in such settings. As the results suggest, the variance parameter in the model can be learnt and the the weights of prediction from the classifiers and unsupervised information can be adjusted to refine the predictions.

Chapter 4

Proposed Work

We conceptually outline four different lines of future work:

- Active multitask learning using annotators' rationale.
- Knowledge transfer across corpora of different languages for category classification.
- Large scale multitask learning using topic models.
- Non-parametric dynamic topic models.

The first three works fall under the category of concurrent knowledge transfer. The fourth work belongs to the category of continual knowledge transfer.

4.1 Active Multitask Learning using Annotators' Rationale

Given the success of active multitask learning based frameworks, further improvement in predictive modeling can be done using extra feedback from the annotators. Instead of providing the correct answers regarding which document should belong to which class and which document should have which set of supervised topics, the annotators can also provide some rationale explaining their answers. For example, if an annotator believes that a document should belong to some class, then they can also highlight parts of the text or set of words that influenced him/her to make that decision. Similarly, for an image, the annotator can provide some explanation by marking the relevant part of the image that made them think to make the corresponding decision.

Earlier works (Zaidan et al., 2008; Donahue and Grauman, 2011), as pointed out in Section 2.4, build on a soft max-margin formulation where the features, as found relevant by the annotators, are removed from the examples. Also, these formulations are valid only for binary classification problems. Once some features are removed from the positive examples, they are assumed to be sort of negative examples in the sense that the decision hyperplane is forced to separate them from the unperturbed positive examples. Our application involves a multi-class problem and also we require the annotators to give feedback on i) for selecting the class label and ii) for selecting the supervised topics. Below, we propose a framework, built on Act-DSLDA, which can accept annotators' rationale and potentially improve the predictive performance when labeled data is hard to come by.

Assume we are given an initial training corpus $\mathcal{L}$ with N documents belonging to Y different classes (where each document belongs to exactly one class and each class corresponds to a different task). Further assume that each of these training documents is also annotated with a set of K_2 different "supervised topics". The objective is to train a model using the words in a data, as well as the associated supervised topics and class labels, and then use this model to classify completely unlabeled test data for which no topics nor class labels are provided. When the learning starts, $\mathcal{L}$ is assumed to have fully labeled documents. However, as

the learning progresses more documents are added to the pool $\mathcal{L}$ with class and/or a subset of supervised topics labeled. Therefore, at any intermediate point of the learning process, $\mathcal{L}$ can be assumed to contain several sets: $\mathcal{L} = \{\mathcal{T} \cup \mathcal{T}_C \cup \mathcal{T}_{A_1} \cup \mathcal{T}_{A_2} \cup \cdots \cup \mathcal{T}_{A_{K_2}}\}$, where $\mathcal{T}$ contains fully labeled documents (*i.e.* with both class and all of supervised topics labeled) and $\mathcal{T}_C$ represents the documents that have class labels. For $1 \leq k \leq K_2$, $\mathcal{T}_{A_k}$ represents the documents that have the k^{th} supervised topic labeled. Since, human provided annotations and class labels are expensive to obtain, we design an active learning framework where the model can query over an unlabeled pool $\mathcal{U}$ and request either class labels or a subset of the supervised topics.

Let the n^{th} document be annotated by L_n annotators. The l_n^{th} annotation comes with an explanation of the form of highlighting set of words $\{w'_{nl_n}\}_{l_n}$ which the corresponding annotator found relevant for identifying the document's class label or supervised topic. Once the set of words $\{w'_{nl}\}_{l_n}$ is removed from the document, it no longer belongs to the class that the annotator identified. However, it does not also mean that the derived document can belong to some other class. To avoid such ambiguity for such derived documents, we define one extra class which we call a "negative" class, index it by $(Y+1)$, and assign the derived documents to this particular class. In case of rationales for supervised topics, only the set of words can be removed and the derived document is not assumed to be annotated by that supervised topic anymore. The corresponding attribute classifier can also be retrained with this derived attribute as done in normal annotator rationale work (explained in Section 2.4).

• For the n^{th} document, sample a topic selection probability vector $\boldsymbol{\theta}_n \sim \text{Dir}(\boldsymbol{\alpha}_n)$, where $\boldsymbol{\alpha}_n = \boldsymbol{\Lambda}_n \boldsymbol{\alpha}$ and $\boldsymbol{\alpha}$ is the parameter of a Dirichlet distribution of dimension K, which is the total number of topics. The topics are assumed to be of two types – latent and supervised, and there are K_1 latent topics and K_2 supervised topics. Therefore, $K = K_1 + K_2$. Latent topics are never observed, while supervised topics are observed in training but not in test data. Henceforth, in each vector or matrix with K components, it is assumed that the first K_1 components correspond to the latent topics and the next K_2 components to the supervised topics. $\boldsymbol{\Lambda}_n$ is a diagonal binary matrix of dimension $K \times K$. The k^{th} diagonal entry is unity if *either* $1 \leq k \leq K_1$ *or* $K_1 < k \leq K$ and the n^{th} document is tagged with the k^{th} topic. Also, $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2)$ where $\boldsymbol{\alpha}_1$ is a parameter of a Dirichlet distribution of dimension K_1 and $\boldsymbol{\alpha}_2$ is a parameter of a Dirichlet distribution of dimension K_2.

• For the m^{th} word in the n^{th} document, sample a topic $z_{nm} \sim \text{multinomial}(\boldsymbol{\theta}'_n)$, where $\boldsymbol{\theta}'_n = (1-\epsilon)\{\boldsymbol{\theta}_{nk}\}_{k=1}^{k_1} \epsilon \{\Lambda_{n,kk}\boldsymbol{\theta}_{nk}\}_{k=1+k_1}^{K}$. This implies that the supervised topics are weighted by ϵ and the latent topics are weighted by $(1-\epsilon)$. Sample the word $w_{nm} \sim \text{multinomial}(\boldsymbol{\beta}_{z_{nm}})$, where $\boldsymbol{\beta}_k$ is a multinomial distribution over the vocabulary of words corresponding to the k^{th} topic.

• For the n^{th} document, generate $Y_n = \arg\max_y \boldsymbol{r}_y^T \mathbb{E}(\bar{\mathbf{z}}_n)$ where Y_n is the class label associated with the n^{th} document, $\bar{\mathbf{z}}_n = \sum_{m=1}^{M_n} \mathbf{z}_{nm}/M_n$. Here, $\mathbf{z}_{nm}$ is an indicator vector of dimension K. $\boldsymbol{r}_y$ is a K-dimensional real vector corresponding to the y^{th} class, and it is assumed to have a prior distribution $\mathcal{N}(0, 1/C)$. M_n is the number of words in the n^{th} document. The maximization problem to generate Y_n (or the classification problem) is carried out using a max-margin principle similar to MedLDA.

• For the l_n^{th} document derived by removing words from the n^{th} document, the generative process is same and limited to the set of words contained in that derived document. If the annotator's feedback on this derived document is for the class label, the derived document is included in the set $\mathcal{C}_n$. Any document that belongs to this set is assigned to the $(Y+1)^{\text{th}}$ class during training.

The optimization problem after variational approximation takes the following form:

$$\min_{\{\boldsymbol{\kappa}_n^v\},\{\boldsymbol{\kappa}_{nl_n}^v\},\boldsymbol{\kappa},\{\xi_n\},\{\xi_{nl_n}\}} \frac{1}{2}||\boldsymbol{r}||^2 - \sum_{n=1}^{N} \ell(\boldsymbol{\kappa}_n^v, \boldsymbol{\kappa}) - \sum_{n=1}^{N}\sum_{l_n=1}^{L_n} \ell(\boldsymbol{\kappa}_{nl_n}^v, \boldsymbol{\kappa}) + C_1 \sum_{n=1}^{N} \xi_n + C_2 \sum_{n=1}^{N}\sum_{l_n=1}^{L_n} \xi_{nl_n}, \quad (4.1)$$

$$\text{s.t. } \forall n \in \mathcal{T}_C, y \in \{1,2,\cdots,Y\} \setminus \{Y_n\} : \mathbb{E}[\boldsymbol{r}^T \Delta f_n(y)] \geq (1-\xi_n); \xi_n \geq 0.$$

$$\text{s.t. } \forall l_n \in \{\mathcal{T}_C \setminus \mathcal{C}_n\}, y \in \{1,2,\cdots,Y\} \setminus \{Y_n\} : \mathbb{E}[\boldsymbol{r}^T \Delta f_{nl_n}(y)] \geq (1-\xi_{nl_n}); \xi_{nl_n} \geq 0.$$

$$\text{s.t. } \forall l_n \in \{\mathcal{T}_C \cap \mathcal{C}_n\}, y \in \{1,2,\cdots,Y\} : \mathbb{E}[\boldsymbol{r}^T \Delta f_{nl_n}(y)] \geq \mu(1-\xi_{nl_n}); \xi_{nl_n} \geq 0.$$

4.2 Knowledge Transfer across Corpora of Different Languages for Category Classification

In Yahoo category taxonomy (YCT), there are lot of labeled data available in English. However, not much of labeled information is available for other languages like German, Spanish, Italian and French. The classification of web pages become challenging in such cases. For example, Table 4.2 demonstrates the amount of labeled information available for the top few categories in YCT. The category names have been anonymized for privacy issues.

Category Number	English	French	Spanish	Italian	German
1	7360	63	408	493	112
2	5162	65	286	258	71
3	4021	102	226	173	42
4	4838	85	184	207	57
5	6070	70	168	210	27
6	1894	38	167	118	21
7	4163	18	138	124	60

One could propose numerous models to solve the problem of data sparsity by transferring knowledge from English to other corpora. However, the proposed model needs to be scalable enough for large scale implementations. Therefore, any model that involves machine translation should be avoided for computational bottleneck. We, therefore, propose an architecture displayed in Fig. 4.1. On a high level, the documents in language other than English are first translated using a dictionary. This is a word-to-word translation and computationally cheap. We define a vocabulary consisting of unique words appearing across all the corpora. The documents in English and other foreign languages are then projected onto the same topic space. The topic level representation of the documents are then used as features for predicting the class labels of the documents using max-margin principle. This method is different from the "projection" method (Yarowsky et al., 2001) suggested in machine translation domain.

The Hierarchical Dirichlet Process and its alternative stick breaking construction proposed in (Wang et al., 2011) are used for the model **N**on-parametric **S**upervised **L**atent **D**irichlet **A**llocation for **M**ultiple **C**orpora (**NP-SLDAMC**). The model is described below:

- Sample $\beta'_k \sim \text{Beta}(1,\gamma)\ \forall k \in \{1,2,\cdots,\infty\}$.
- Set $\beta_k = \beta'_k \prod_{j=1}^{k-1}(1-\beta'_j)\ \forall k \in \{1,2,\cdots,\infty\}$.
- For the n^{th} document in the g^{th} corpus (we assume that there are G corpora and the g^{th} corpus has N_g documents),
  - Sample $\psi_{gnt} \sim G_0\ \forall t \in \{1,2,\cdots,\infty\}$.
  - Sample $\pi'_{gnt} \sim \text{Beta}(1,\alpha)\ \forall t \in \{1,2,\cdots,\infty\}$.
  - Set $\pi_{gnt} = \pi'_{gnt} \prod_{l=1}^{t-1}(1-\pi'_{gnl})$.

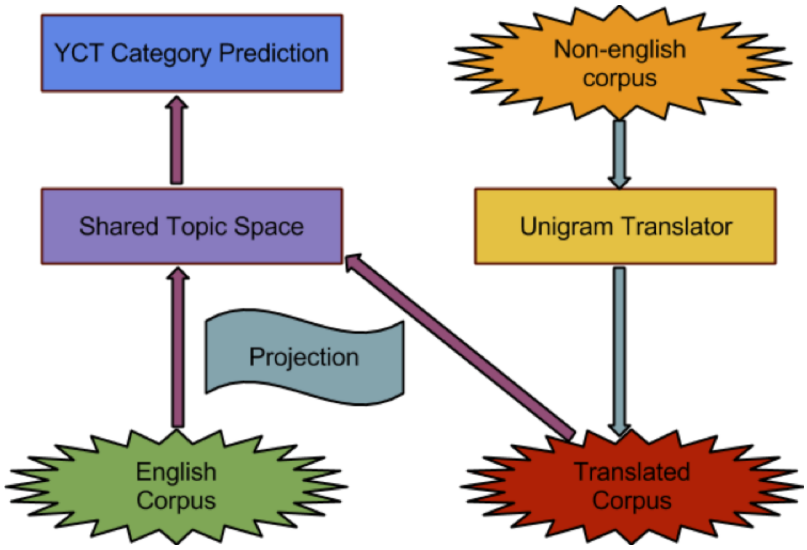

Figure 4.1: System Overview

- For the m^{th} word in the n^{th} document from the g^{th} corpus,
  - Sample $z_{gnm} \sim \text{multinomial}(\boldsymbol{\pi}_{gn})$.
  - Sample $w_{gnm} \sim \text{multinomial}(\psi_{nz_{gnm}})$.
- For the n^{th} document in the g^{th} corpus, generate $Y_{gn} = \arg\max_y \boldsymbol{r}_{gy}^T \mathbb{E}(\bar{\mathbf{z}}_{gn})$ where Y_{gn} is the class label associated with the corresponding document, $\bar{\mathbf{z}}_{gn} = \sum_{m=1}^{M_n} \mathbf{z}_{gnm}/M_{gn}$. $\boldsymbol{r}_{gy}$ is an infinite dimensional real vector corresponding to the g^{th} corpus and y^{th} class, and it is assumed to have a prior distribution $\mathcal{N}(0, 1/C)$. M_{gn} is the number of words in the n^{th} document from the g^{th} corpus. The maximization problem to generate Y_{gn} (or the classification problem) is carried out using a max-margin principle and variational approximation similar to NP-DSLDA.

In the proposed model shown in Fg. 4.2, a draw from a top-level DP yields the base measures for DPs associated with the documents of all the different corpora. Draws from the document-level DPs provide a representation of each document as a probability distribution across topics. Thus, the model allows topics to be shared across corpora. The proposed model will be compared to baselines that involve computationally complex machine translation algorithms and also to algorithms that do not involve translation at all and just treat each corpus separately.

One can further implement online inference methods to update the parameters with the arrival of new labeled documents. Active learning for efficient query over unlabeled pool can also be implemented for the non-English corpora.

4.3 Large Scale Multitask Learning using Topic Models

In many large scale applications, the number of data points as well as the feature dimension are both extremely large. For example, one can consider a corpus of text documents where the number of documents is huge and the size of the vocabulary is of the order of several tens of thousands. Additionally, the documents might belong to some groups. Such groups can be discovered in a completely unsupervised fashion or there might also be partial supervision available for some of these groups.

Consider a scenario where there are millions of users (or tasks) and each user has a set of documents. The problem starts with a task by feature matrix where the feature dimension is the collection of unique words

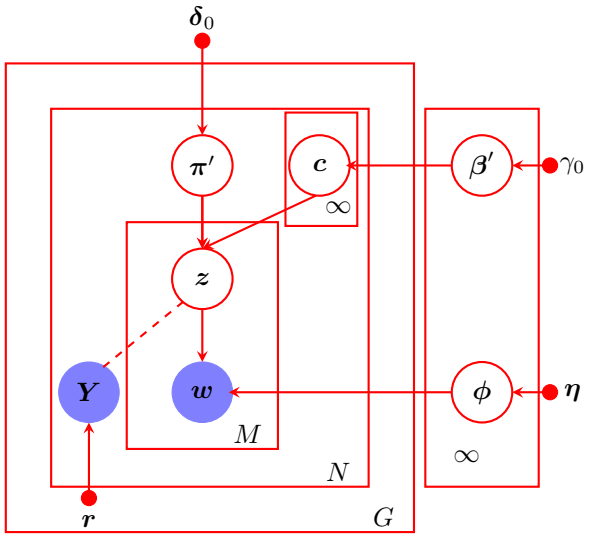

Figure 4.2: NP-SLDAMC

appearing in all the documents of all the users. Since there are millions of users (or tasks) and thousands of features, it makes sense to assume some low dimensional representation of both rows and columns to make any useful prediction or discover underlying structure of the data. Therefore, the idea here is to decompose this matrix as a product of three matrices – tasks by task clusters, tasks by feature clusters and feature clusters by features. In the first matrix, the tasks are categorized into some groups. Tasks in each group are assumed to have a distinctive distribution in the space of feature clusters, as dictated by the second matrix. The third matrix denotes a grouping of the features. Therefore, the rows and the columns of the original matrix are both clustered and if they are done in an interactive way, this is an instance of co-clustering (Banerjee et al., 2007; Deodhar and Ghosh, 2007; Ding et al., 2006). The scheme is shown in Fig. 4.3.

The intermediate matrix in such decomposition has unobserved latent variables in both dimensions. This might create problem in the optimization framework for identifiability issues. Therefore, this intermediate matrix is decomposed as a product of two matrices – task groups by documents and documents by feature clusters. This scheme is shown in Fig. 4.4.

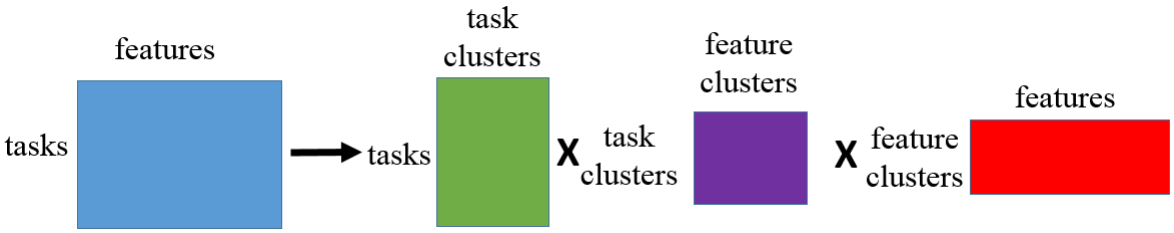

Figure 4.3: First Scheme of Decomposition

One can now see an NMF framework if the last two matrices are considered. The decomposition of the last two matrices will be solved using a topic model framework where the features are the word unigrams and the feature clusters are topics. The third matrix will give an assignment of the documents in the topic space and the fourth matrix will contain the topic structure.

One could solve the problem of task grouping and topic modeling in a separate framework. However, as exemplified in a recent research on integrating document clustering and topic modeling (Xie and Xing, 2013), incorporating information about documents' latent groups further improves the performance of topic

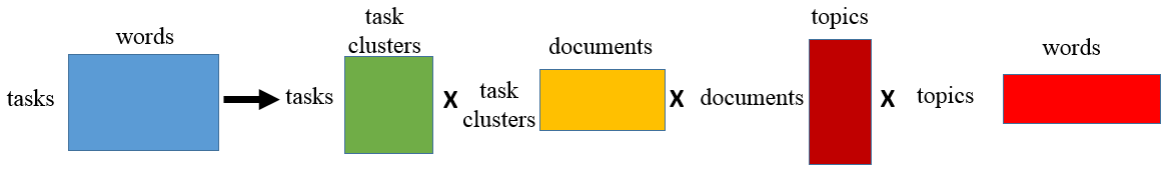

Figure 4.4: Second Scheme of Decomposition

models and the assignment of the documents into latent groups can be bettered from this revised topic model. Our objective is to follow this research direction and propose methods for large scale distributed inference techniques and possibly extend the same to non-parametric setting where the number of groups can be inferred from the data itself. For large scale applications, online variational inference has been shown to be successful (Hoffman et al., 2010a; Wang et al., 2011), as illustrated in Section 2.9.2. Therefore, such framework is planned to be used for large scale non-parametric multitask learning.

Much of the large scale data available in industry reside in different machines and necessitates distributed processing instead of aggregating all the data in the same place. One particular scenario is when the features of the data lie in different physical locations. For example, the data for Yahoo! mail and Yahoo! news for the same user reside in different cluster machines. Techniques for distributed online variational inference using alternating direction method of multipliers (ADMM)(Boyd et al., 2011) will also be proposed.

4.4 Non-parametric Dynamic Topic Models

As explained in Section 2.10, no existing work systematically deals with a non-parametric extension of dynamic topic model that can adapt the prediction with changes in the data that brings about new topics or modifications in existing topics. To address this problem, an NB process proposed in Zhou and Carin (2013) to model the random count vectors at time t as $\boldsymbol{X}_{tj} \sim \text{NBP}(G_t, p_{tj})$, where $j = 1, 2, \cdots, J_t$ is the group index for groups at time t. To model the evolution of these count vectors, a Markov structure is further imposed between successive gamma processes across time as $G_t \sim \text{GaP}(c, G_{t-1})$, where G_0 is a continuous base measure with a finite total mass $\gamma_0 = G_0(\Omega)$. This leads to a dynamic gamma-negative binomial process (GNBP) as

$$\boldsymbol{X}_{tj} \sim \text{NBP}(G_t, p_{tj}), G_t \sim \text{GaP}(c, G_{(t-1)}) : \tag{4.2}$$

where $t = 1, 2, \cdots, T$ is the time index. In the GNBP all the groups share the same gamma process, whereas in the dynamic GNBP the groups $(\boldsymbol{X}_{t_1}, \cdots, \boldsymbol{X}_{tJ_t})$ at time t share a same gamma process G_t, whose base measure is $G_{(t-1)}$. Groups belonging to the same time are assumed to be exchangeable. This dynamic construction allows sharing statistical strengthen across time. The main challenge for this model is the inference of G_t, which is connected to $G_{(t-1)}$, $G_{(t+1)}$ and $\boldsymbol{X}_{tj}$. Using the results from Zhou and Carin (2013), two equivalent augmentations for the dynamic GNBP can be presented:

$$L_{tj} \sim \text{CRTP}(\boldsymbol{X}_{tj}, G_t); \boldsymbol{X}_{tj} \sim \text{NBP}(G_t, p_{tj}) \tag{4.3}$$

$$\boldsymbol{X}_{tj} \sim \sum_{\ell=1}^{L_{tj}} Log(p_{tj}), L_{tj} \sim \text{PP}(-G_t ln(1 - p_{tj})) \tag{4.4}$$

Following Zhou and Carin (2013), two more equivalent augmentations can be presented:

$$L'_t \sim \text{CRTP}(L_t, G_{(t-1)}), L_t \sim \text{NBP}(G_{(t-1)}, p'_t) \tag{4.5}$$

$$L_t \sim \sum_{\ell=1}^{L'_t} Log(p'_t), L'_t \sim \text{PP}(-G_{(t-1)} ln(1 - p'_t)) \tag{4.6}$$

where

$$L_t = L'_{(t+1)} + \sum_{j=1}^{J_t} L_{tj} \tag{4.7}$$

and

$$p'_t = \frac{-ln(1-p'_{(t+1)}) - \sum_{j=1}^{J_t} ln(1-p_{tj})}{c - ln(1-p'_{(t+1)}) - \sum_{j=1}^{J_t} ln(1-p_{tj})} \tag{4.8}$$

with $p_{(T+1)j} = 0, p_{0j} = 0, L_{(T+1)} = 0, L'_{(T+1)} = 0$ and $L_0 = L'_1$.

A GNBP dynamic mixed-membership model is further constructed as follows:

$$\begin{aligned} x_{tji} \sim F(\omega_{z_{tji}}), \omega \sim g_0, z_{tji} \sim \sum_{k=1}^{\infty} \frac{\theta_{tjk}}{\Theta_{tj}}(\Omega)\delta_k, m_{tj} \sim \text{Pois}(\Theta_{tj}(\Omega)), \\ \Theta_{tj} \sim GaP((1-p_{tj})/ptj, G_t), G_t \sim \text{GaP}(c, G_{(t-1)}) \end{aligned} \tag{4.9}$$

In this model, each group is modeled with a random count vector $(n_{tj1}, \cdots, n_{tj\infty})$ in the latent space, where $n_{tjk} = \sum_{j=1}^{J_t} \delta_{z_{tji}=k}$. Further time-evolving topics are considered that are generated as

$$\phi_k^{(t)} \sim Dir(\eta V \phi_k^{(t-1)}) \tag{4.10}$$

where η is a smoothing parameter and $\phi_k^{(0)} = (1/V, \cdots, 1/V)^T$.

Non-parametric dynamic topic model will further be extended to model the dynamics of documents as well the users associated with the documents. These users can be the readers of the documents or the authors and it is of interest to industry to track the evolution of topics as the social networks of the users evolve over time. Fu et al. (2009) introduced a dynamic stochastic block model which can model the evolution of social network. This models will be integrated with non-parametric dynamic topic model.

Chapter 5

Datasets for Empirical Evaluation

So far, the evaluation has used two datasets, a multi-class image database aYahoo and a text corpus with documents from conferences. For the proposed works in Section 4.2 and 4.3, data from Yahoo! Inc. is going to be used which cannot be described here for privacy and copyright issues. For the proposed work in Section 4.1, the conference dataset will be annotated on mechanical turk by human experts. For non-parametric dynamic topic models, Science data will be used, as described below.

5.1 aYahoo Data

aYahoo image dataset (Farhadi et al., 2009) has 12 classes – carriage, centaur, bag, building, donkey, goat, jetski, monkey, mug, statue, wolf, and zebra `http://vision.cs.uiuc.edu/attributes/`. Each image is annotated with relevant visual attributes such as "has head", "has wheel", "has torso" and 61 others, which we use as the supervised topics. Using such intermediate "attributes" to aid visual classification has become a popular approach in computer vision (Lampert et al., 2009; Kovashka et al., 2011). After extracting SIFT features (Lowe, 2004) from the raw images, quantization into 250 clusters is performed, defining the vocabulary for the bag of visual words. Images with less than two attributes were discarded.

5.2 ACM Conference Data

The text corpus consists of conference paper abstracts from two groups of conferences. The first group has four conferences related to data mining – WWW, SIGIR, KDD, and ICML, and the second group consists of two VLSI conferences – ISPD and DAC. The classification task is to determine the conference at which the abstract was published. As supervised topics, we use keywords provided by the authors, which are presumably useful in determining the conference venue. Since authors usually take great care in choosing keywords so that their paper is retrieved by relevant searches, we believed that such keywords made a good choice of supervised topics. Part of the data, crawled from ACM's website, was used in Wang et al. (2009). A total of 2,300 abstracts were collected each of which had at least three keywords and an average of 78 ($\pm$33.5) words. After stop-word removal, the vocabulary size for the assembled data is 13,412 words. The final number of supervised topics, after some standard pre-processing of keywords, is 55.

5.3 Science Data

This dataset, used in Blei and Lafferty (2006), contains 30,000 articles from Science, 250 from each of the 120 years between 1881 and 1999. The data was collected by JSTOR (www.jstor.org), a notfor-profit organization that maintains an online scholarly archive obtained by running an optical character recognition (OCR) engine over the original printed journals. JSTOR indexes the resulting text and provides online

access to the scanned images of the original content through keyword search. The corpus is made up of approximately 7.5 million words. After standard text pre-processing, the total vocabulary size is 15,955.

Bibliography

A. Acharya, E. R. Hruschka, J. Ghosh, B. Sarwar, and J.D. Ruvini. Probabilistic combination of classifier and cluster ensembles for non-transductive learning. In *SDM*, 2013a.

A. Acharya, A. Rawal, R. J. Mooney, and E. R. Hruschka. Using both supervised and latent shared topics for multitask learning. In *ECML PKDD, Part II, LNAI 8189*, pages 369–384, 2013b.

A. Acharya, J. Ghosh, and R. J. Mooney. Active multitask learning using both latent and supervised shared topics. In *Submitted to SDM*, 2014.

A. Ahmed and E. P. Xing. Dynamic non-parametric mixture models and the recurrent chinese restaurant process: with applications to evolutionary clustering. In *Proc. of SDM*, pages 219–230, 2008.

R. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005. ISSN 1532-4435.

R. K. Ando. Applying alternating structure optimization to word sense disambiguation. In *Proceedings of Computational Natural Language Learning*, 2006.

C. E. Antoniak. Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems. *The Annals of Statistics*, 2:1152–1174, 1974.

A. Argyriou, C. A. Micchelli, M. Pontil, and Y. Ying. A spectral regularization framework for multi-task structure learning. In *Proc. of NIPS*, 2007.

A. Asuncion, M. Welling, P. Smyth, and Y. W. Teh. On smoothing and inference for topic models. In *Proc. of UAI*, pages 27–34, 2009.

B. Bakker and T. Heskes. Task clustering and gating for Bayesian multitask learning. *Journal of Machine Learning Research*, 4, 2003.

A. Banerjee, I. Dhillon, J. Ghosh, S. Merugu, and D. Modha. A generalized maximum entropy to bregman co-clustering and matrix approximation. *Journal of Machine Learning Research*, 8:1919–1986, 2007.

S. Ben-David and R. Schuller. Exploiting task relatedness for multiple task learning. In *Learning Theory and Kernel Machines*, volume 2777, pages 567–580. 2003.

S. Bickel, J. Bogojeska, T. Lengauer, and T. Scheffer. Multi-task learning for HIV therapy screening. In *Proceedings of International Conference on Machine Learning*, pages 56–63, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4.

I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987.

D. Blackwell and J. B. Macqueen. Ferguson distributions via Pólya urn schemes. *The Annals of Statistics*, 1:353–355, 1973.

D. M. Blei and J. D. Lafferty. Dynamic topic models. In *Proc. of ICML*, pages 113–120, 2006.

D. M. Blei and J. D. Mcauliffe. Supervised topic models. In *Proc. of NIPS*, 2007.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *JMLR*, 3:993–1022, 2003.

K. D. Bollacker and J. Ghosh. Knowledge transfer mechanisms for characterizing image datasets. In *Soft Computing and Image Processing*. Physica-Verlag, Heidelberg, 2000.

A. Bordes, S. Ertekin, J. Weston, and L. Bottou. Fast kernel classifiers with online and active learning. *JMLR*, 6:1579–1619, December 2005.

A. Bordes, L. Bottou, P. Gallinari, and J. Weston. Solving multiclass support vector machines with larank. In *Proc. of ICML*, pages 89–96, 2007.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. Tech Report, 2011.

W. L. Buntine. Variational extensions to em and multinomial pca. In *Proc. of ECML*, pages 23–34, 2002.

R. Caruana. Multitask learning. *Machine Learning*, 28:41–75, July 1997.

A. T. Cemgil. Bayesian inference for nonnegative matrix factorisation models. *Intell. Neuroscience*, 2009: 4:1–4:17, January 2009.

N. Cesa-Bianchi, C. Gentile, and L. Zaniboni. Worst-case analysis of selective sampling for linear classification. *JMLR*, 7:1205–1230, 2006.

Y. S. Chan and H. T. Ng. Domain adaptation with active learning for word sense disambiguation. In *Proc. of ACL*, pages 49–56, 2007.

J. Chang and D. Blei. Relational topic models for document networks. In *Proc. of AISTATS*, 2009.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

M. Deodhar and J. Ghosh. A framework for simultaneous co-clustering and learning from complex data. In *Knowledge Discovery and Data Mining '07*, pages 250–259, 2007.

C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proc. of KDD*, pages 126–135, 2006.

J. Donahue and K. Grauman. Annotator rationales for visual recognition. In *Proc. of ICCV*, pages 1395–1402, 2011.

M. D. Escobar and M. West. Bayesian Density Estimation and Inference Using Mixtures. *Journal of the American Statistical Association*, 90(430):577–588, June 1995.

T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005. ISSN 1532-4435.

T. Evgeniou, M. Pontil, and O. Toubia. A convex optimization approach to modeling consumer heterogeneity in conjoint estimation. *Marketing Science*, 26(6):805–818, November 2007.

A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *Proc. of CVPR*, pages 1778–1785, 2009.

W. Fu, L. Song, and E. P. Xing. Dynamic mixed membership blockmodel for evolving networks. In *Proc. of ICML*, pages 329–336, 2009.

M. Greenwood and G. U. Yule. An Inquiry into the Nature of Frequency Distributions Representative of Multiple Happenings with Particular Reference to the Occurrence of Multiple Attacks of Disease or of Repeated Accidents. *J. Roy. Stat. Soc.*, 83, 1920.

A. Harpale and Y. Yang. Active learning for multi-task adaptive filtering. In *Proc. of ICML*, pages 431–438. Omnipress, 2010.

M. D. Hoffman, D. M. Blei, and F. Bach. Online learning for latent dirichlet allocation. In *Proc. of NIPS*, 2010a.

M. D. Hoffman, D. M. Blei, and P. R. Cook. Bayesian nonparametric matrix factorization for recorded music. In *Proc. of ICML*, pages 439–446, 2010b.

P. A. D. F. R. Højen-Sørensen, O. Winther, and L. K. Hansen. Mean-field approaches to independent component analysis. *Neural Comput.*, 14:889–918, 2002.

L. Jacob, F. Bach, and J.-P. Vert. Clustered multi-task learning: A convex formulation. *CoRR*, abs/0809.2085, 2008.

P. Jain and A. Kapoor. Active learning for large multi-class problems. In *Proc. of CVPR*, pages 762–769, 2009.

R. Jenatton, J. Audibert, and F. Bach. Structured variable selection with sparsity-inducing norms. *JMLR*, 12:2777–2824, nov 2011.

M. I. Jordan. Hierarchical models, nested models, and completely random measures. In M.-H. Chen, D. Dey, P. Muller, D. Sun, and K. Ye, editors, *Frontiers of Statistical Decision Making and Bayesian Analysis: In Honor of James O. Berger*, pages 207–217. Springer, 2010.

A.J. Joshi, F. Porikli, and N. Papanikolopoulos. Multi-class active learning for image classification. In *Proc. of CVPR*, pages 2372–2379, 2009.

G. Jun and J. Ghosh. An efficient active learning algorithm with knowledge transfer for hyperspectral remote sensing data. In *Proc. of International Geosci. and Sens. Symposium*, volume 1, pages I–52–I–55, 2008.

R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME Journal of Basic Engineering*, (82 (Series D)):35–45, 1960.

S. Kim and E. P. Xing. Tree-guided group lasso for multi-task regression with structured sparsity. In *Proc. of ICML*, pages 543–550, 2010.

J. F. C. Kingman. Completely random measures. *Pacific J. Math*, 21(1):59–78, 1967.

A. Kovashka, S. Vijayanarasimhan, and K. Grauman. Actively selecting annotations among objects and attributes. In *Proc. of ICCV*, pages 1403–1410, 2011.

C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by betweenclass attribute transfer. In *Proc. of CVPR*, pages 951–958, 2009.

D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *In Neural Information Processing Systems*, pages 556–562. MIT Press, 2000.

Y. Low, D. Agarwal, and A. J. Smola. Multiple domain user personalization. In *Proc. of KDD*, pages 123–131, 2011.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

J. W. Miskin. Ensemble learning for independent component analysis. Technical report, in Advances in Independent Component Analysis, 2000.

R. M. Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal OF Computational and Graphical Statistics*, 9(2):249–265, 2000.

R. M. Neal and G. E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants, 1999.

P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.

S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22:1345–1359, 2010.

A. Passos, P. Rai, J. Wainer, and H. Daumé III. Flexible modeling of latent task structures in multitask learning. In *Proc. of ICML*, pages 1103–1110, 2012.

J. Pitman. *Combinatorial Stochastic Processes: Preliminaries*, pages 1–11. Springer-Verlag, 2006.

M.D. Plumbley and E. Oja. A "nonnegative pca" algorithm for independent component analysis. *Neural Networks, IEEE Transactions on*, 15(1):66–76, 2004.

G.J. Qi, Xian-Sheng H., Yong R., Jinhui T., and Hong-Jiang Z. Two-dimensional active learning for image classification. In *Proc. of CVPR*, pages 1–8, 2008.

A. Quattoni, S. Wang, L. P Morency, M. Collins, and T. Darrell. Hidden-state conditional random fields. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007.

P. Rai, A. Saha, H. Daumé, III, and S. Venkatasubramanian. Domain adaptation meets active learning. In *Proc. of NAACL HLT Workshop on Active Learning for Natural Language Processing*, pages 27–32, 2010.

D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora. In *Proc. of EMNLP*, pages 248–256, 2009.

S. Robertson and I. Soboroff. The trec 2002 filtering track report. In *Text Retrieval Conference*, 2002.

N. Roy and A. K. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proc. of ICML*, pages 441–448, 2001.

T. N. Rubin, A. Chambers, P. Smyth, and M. Steyvers. Statistical topic models for multi-label document classification. CoRR, abs/1107.2462, 2011.

A. Saha, P. Rai, H. Daum III, and S. Venkatasubramanian. Online learning of multiple tasks and their relationships. *JMLR - Proceedings Track*, 15:643–651, 2011.

J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

D. Shen, J. Ruvini, and B. Sarwar. Large-scale item categorization for e-commerce. In *CIKM*, Oct. 2012.

X. Shi, W. Fan, and J. Ren. Actively transfer domain knowledge. In *Proc. of ECML PKDD - Part II*, pages 342–357, 2008.

N. D. Socci, D. D. Lee, and H. S. Seung. The rectified gaussian distribution. In *Proc. of NIPS*, pages 350–356, 1998.

Y. W. Teh. Dirichlet process. pages 280–287, 2010.

Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, 101:1566–1581, December 2006.

Y. W. Teh, K. Kurihara, and M. Welling. Collapsed variational inference for HDP. In *Proc of NIPS*, 2007.

A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(5):854–869, May 2007.

C. Wang, D. M. Blei, and D. Heckerman. Continuous time dynamic topic models. In *Proc. of UAI*, pages 579–586, 2008.

C. Wang, B. Thiesson, C. Meek, and D. Blei. Markov topic models. In *Proceedings of Artificial Intelligence and Statistics*, 2009.

C. Wang, J. W. Paisley, and D. M. Blei. Online variational inference for the hierarchical Dirichlet process. *JMLR - Proceedings Track*, 15:752–760, 2011.

Y. Wang and L. Carin. Levy measure decompositions for the beta and gamma processes. In *Proc. of ICML*, 2012.

Y. Wang, S. X. Liu, J. Feng, and L. Zhou. Mining Naturally Smooth Evolution of Clusters from Dynamic Data. In *Proc. of SDM*, 2007.

L. Wasserman. *All of Nonparametric Statistics (Springer Texts in Statistics)*. Springer, 2005.

K. Weinberger, A. Dasgupta, J. Langford, A. Smola, and J. Attenberg. Feature hashing for large scale multitask learning. In *Proc. of ICML*, pages 1113–1120, 2009.

R. L. Wolpert, M. A. Clyde, and C. Tu. Stochastic expansions using continuous dictionaries: Lévy adaptive regression kernels. *The Annals of Statistics*, 39(4):1916–1962, December 2011.

P. Xie and E.P. Xing. Integrating document clustering and topic modeling. In *Proc. of UAI*, pages 431–438, 2013.

Y. Xue, X. Liao, L. Carin, and B. Krishnapuram. Multi-task learning for classification with Dirichlet process priors. *Journal of Machine Learning Research*, 8:35–63, 2007. ISSN 1532-4435.

D. Yarowsky, G. Ngai, and R. Wicentowski. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of HLT 2001, First International Conference on Human Language Technology Research*, 2001.

O. F. Zaidan, J. Eisner, and C. Piatko. Machine learning with annotator rationales to reduce annotation cost. In *Proc. of the NIPS Workshop on Cost Sensitive Learning*, 2008.

J. Zhang, Z. Ghahramani, and Y. Yang. Flexible latent variable models for multi-task learning. *Machine Learning*, 73(3):221–242, December 2008.

M. Zhou and L. Carin. Negative binomial process count and mixture modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99(PrePrints), 2013.

J. Zhu, A. Ahmed, and E. P. Xing. MedLDA: maximum margin supervised topic models for regression and classification. In *Proc. of ICML*, pages 1257–1264, 2009.

Appendix A

Publications

1. Acharya, Ayan, Hruschka, Eduardo R., Ghosh, Joydeep, Sarwar, Badrul, and Ruvini, Jean-David, Probabilistic Combination of Classifier and Cluster Ensembles for Non-transductive Learning, SDM, 2013 [.pdf].

2. Gunasekar, Suriya, Acharya, Ayan, Gaur, Neeraj, and Ghosh, Joydeep, Noisy Matrix Completion Using Alternating Minimization, ECML PKDD, Part II, LNAI 8189, pp.194-209, 2013 [.pdf].

3. Acharya, Ayan, Rawal, Aditya, Mooney, Raymond J., and Hruschka, Eduardo R., Using Both Supervised and Latent Shared Topics for Multitask Learning, ECML PKDD, Part II, LNAI 8189, pp.369-384, 2013 [.pdf].

4. Ghosh, Joydeep and Acharya, Ayan, Cluster Ensembles: Theory and Applications, in Data Clustering: Algorithms and Applications, 2013 [.pdf].

5. Acharya, Ayan, Mooney, Raymond J., Ghosh, Joydeep, Active Multitask Learning Using Doubly Supervised Latent Dirichlet Allocation, NIPS Topic Model Workshop, 2013 [.pdf].

6. Coletta, Luiz Fernando, Hruschka, Eduardo R., Acharya, Ayan, and Ghosh, Joydeep, Towards the Use of Metaheuristics for Optimizing the Combination of Classifier and Cluster Ensembles, Appearing in 11th Brazilian Congress (CBIC) on Computational Intelligence, 2013 [.pdf].

7. Acharya, Ayan, Hruschka, Eduardo R., Ghosh, Joydeep, and Acharyya, Sreangsu, An Optimization Framework for Combining Ensembles of Classifiers and Clusterers with Applications to Non-transductive Semi-Supervised Learning and Transfer Learning, Appearing in ACM Transactions on Knowledge Discovery from Data, 2013 [.pdf].

8. Ghosh, Joydeep and Acharya, Ayan, A Survey of Consensus Clustering, Appearing in Handbook of Cluster Analysis, 2013 [.pdf].

9. Acharya, Ayan, Hruschka, Eduardo R., Ghosh, Joydeep, and Acharyya, Sreangsu, Transfer Learning with Cluster Ensembles, Journal of Machine Learning Research - Proceedings Track, 27 , pp.123-132, 2012 [.pdf].

10. Acharya, Ayan, Lee, Jangwon, and Chen, An, Real Time Car Detection and Tracking in Mobile Devices, IEEE International Conference on Connected Vehicles and Expo, 2012 [.pdf].

11. Ghosh, Joydeep and Acharya, Ayan, Cluster ensembles, Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery, 1 (4) , pp.305-315, 2011 [.pdf].

12. Acharya, Ayan, Hruschka, Eduardo R., Ghosh, Joydeep, and Acharyya, Sreangsu, C^3E: A Framework for Combining Ensembles of Classifiers and Clusterers, MCS, pp.269-278, 2011 [.pdf].

13. Acharya, Ayan, Hruschka, Eduardo R., and Ghosh, Joydeep, A Privacy-Aware Bayesian Approach for Combining Classifier and Cluster Ensembles, SocialCom/PASSAT, pp.1169-1172, 2011 [.pdf].

Appendix B

Schedule and Vita

Spring 2014 Finish work on non-parametric dynamic topic models and active multitask learning using annotators' rationale.

Summer 2014 Finish work on knowledge transfer across corpora of different languages for category classification.

Fall 2014 Finish work on large scale multitask learning using topic models.

Ayan Acharya received his Bachelor of Engineering degree from the Electronics and Telecommunication Engineering Department, Jadavpur Univesity, Kolkata in 2009. Immediately after that, he joined the Electrical and Computer Engineering Department of University of Texas at Austin in Fall 2009 in the integrated Ph.D. program. He received his Masters degree in May 2012 from the Department of Electrical and Computer Engineering at the University of Texas at Austin. He worked summer interns in eBay Research Lab, Qualcomm Inc., and Yahoo! Labs. He is currently pursuing a Ph.D. at the University of Texas at Austin and is advised by Dr. Joydeep Ghosh and co-advised by Dr. Raymond J. Mooney.

Probabilistic Combination of Classifier and Cluster Ensembles for Non-transductive Learning

Ayan Acharya* Eduardo R. Hruschka† Joydeep Ghosh* Badrul Sarwar‡
Jean-David Ruvini‡

Abstract

Unsupervised models can provide supplementary soft constraints to help classify new target data under the assumption that similar objects in the target set are more likely to share the same class label. Such models can also help detect possible differences between training and target distributions, which is useful in applications where concept drift may take place. This paper describes a Bayesian framework that takes as input class labels from existing classifiers (designed based on labeled data from the source domain), as well as cluster labels from a cluster ensemble operating solely on the target data to be classified, and yields a consensus labeling of the target data. This framework is particularly useful when the statistics of the target data drift or change from those of the training data. We also show that the proposed framework is privacy-aware and allows performing distributed learning when data/models have sharing restrictions. Experiments show that our framework can yield superior results to those provided by applying classifier ensembles only.

1 Introduction

In several data mining applications, one builds an initial classification model that needs to be applied to unlabeled data acquired subsequently. Since the statistics of the underlying phenomena being modeled changes with time, these classifiers may also need to be occasionally rebuilt if performance degrades beyond an acceptable level. In such situations, it is desirable that the classifier functions well with as little labeling of new data as possible, since labeling can be expensive in terms of time and money, and a potentially error-prone process. Moreover, the classifier should be able to adapt to changing statistics to some extent, given the aforementioned constraints.

This paper addresses the problem of combining multiple classifiers and clusterers in a fairly general setting, that includes the scenario sketched above. An ensemble of classifiers is first learnt on an initial labeled training dataset after which the training data can be discarded. Subsequently, when new unlabeled target data is encountered, a cluster ensemble is applied to it, thereby generating cluster labels for the target data. The heart of our approach is a Bayesian framework that combines both sources of information (class/cluster labels) to yield a consensus labeling of the target data.

The setting described above is, in principle, different from transductive learning setups where both labeled and unlabeled data are available at the same time for model building [19], as well as online methods [6]. Additional differences from existing approaches are described in Section 2. For the moment we note that the underlying assumption is that similar new objects in the target set are more likely to share the same class label. Thus, the supplementary constraints provided by the cluster ensemble can be useful for improving the generalization capability of the resulting classifier system. Also, these supplementary constraints can be useful for designing learning methods that help determining differences between training and target distributions, making the overall system more robust against concept drift.

We also show that our approach can combine cluster and classifier ensembles in a privacy-preserving setting. This approach can be useful in a variety of applications. For example, the data sites can represent parties that are a group of banks, with their own sets of customers, who would like to have a better insight into the behavior of the entire customer population without compromising the privacy of their individual customers.

The remainder of the paper is organized as follows. The next section addresses related work. The proposed Bayesian framework — named **BC^{3}E**, from **B**ayesian **C**ombination of **C**lassifiers and **C**lusterer **E**nsembles — is described in Section 3. Issues with privacy preservation are discussed in Section 4 and the experimental results are reported in Section 5. Finally, Section 6 concludes the paper.

2 Related Work

The combination of multiple classifiers to generate an ensemble has been proven to be more useful compared to the use of individual classifiers [17]. Analogously,

*University of Texas at Austin, Austin, TX, USA. Email: {aacharya@, ghosh@ece}.utexas.edu

†University of Sao Paulo at Sao Carlos, Brazil. Email: erh@icmc.usp.br

‡eBay Research Lab, San Jose, CA, USA. Email: {bsarwar, jruvini}@ebay.com

several research efforts have shown that cluster ensembles can improve the quality of results as compared to a single clusterer — *e.g.*, see [21] and references therein. Most of the motivations for combining ensembles of classifiers and clusterers are similar to those that hold for the standalone use of either classifier or cluster ensembles. Additionally, unsupervised models can provide supplementary constraints for classifying new data and thereby improve the generalization capability of the resulting classifier. These successes provide the motivation for designing effective ways of leveraging both classifier and cluster ensembles to solve challenging prediction problems.

Specific mechanisms for combining classification and clustering models however have been introduced only recently in the **B**ipartite **G**raph-based **C**onsensus **M**aximization (**BGCM**) algorithm [13], the **L**ocally **W**eighted **E**nsemble (**LWE**) algorithm [12] and, in the **C^{3}E** algorithm (**C**onsensus between **C**lassification and **C**lustering **E**nsembles – [3]). Both **BGCM** and **C^{3}E** have parameters that control the relative importance of classifiers and clusterers. In traditional semi-supervised settings, such parameters can be optimized via cross-validation. However, if the training and the target distributions are different, cross-validation is not possible. From this viewpoint, our approach (**BC^{3}E**) can be seen as an extension of **C^{3}E** [3] that is capable of dealing with this issue in a more principled way. In addition, the algorithms in [13, 12, 3] do not deal with privacy issues, whereas our probabilistic framework can combine class labels with cluster labels under conditions where sharing of individual records across data sites is not permitted. It uses a soft probabilistic notion of privacy, based on a quantifiable information-theoretic formulation [16]. Note that existing works on Bayesian classifier ensembles ([10, 8, 14]) do not deal with privacy issues.

From the clustering side, the proposed model borrows ideas from the Bayesian Cluster Ensemble [21]. In [1], we introduced some preliminary ideas that are further developed in our current paper. In particular, the algorithm in [1] is not capable of automatically estimating the importance that classifiers and clusterers should have. This property is fundamental for applications where training and target distributions are different. In addition, the Bayesian model presented here is considerably different and requires more sophisticated inference and estimation procedures.

3 Probabilistic Model

We assume that a classifier ensemble has been (previously) induced from a training set. At this point and assuming a non-transductive setting, the training data can be discarded if so desired. Such a classifier ensemble is employed to generate a number of class labels for every object in the target set. **BC^{3}E** refines such classifier prediction with the help of a cluster ensemble. Each base clustering algorithm that is part of the ensemble partitions the target set, providing cluster labels for each of its objects. From this point of view, the cluster ensemble provides supplementary constraints for classifying those objects, with the rationale that similar objects — those that are likely to be clustered together across (most of) the partitions that form the cluster ensemble — are more likely to share the same class label.

Consider a target set $\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$ formed by N unlabeled objects. A classifier ensemble composed of r_1 models has produced r_1 class labels for every object $\mathbf{x}_n \in \mathcal{X}$. It is assumed that the target objects belong to k classes denoted by $C = \{C_i\}_{i=1}^k$ and at least one object from each of these classes was observed in the training phase (*i.e.* we do not consider "novel" classes in the target set). Similarly, consider that a cluster ensemble comprised of r_2 clustering algorithms has generated cluster labels for every object in the target set. The number of clusters need not be the same across different clustering algorithms. Also, it should be noted that the cluster labeled as *1* in a given data partition may not align with the cluster numbered *1* in another partition, and none of these clusters may correspond to class *1*. Given the class and cluster labels, the objective is to come up with refined class probability distributions $\{(\hat{P}(C_i|\mathbf{x}_n))_{i=1}^k = f(\boldsymbol{y}_n)\}_{n=1}^N$ of the target set objects, where $f(\mathbf{y}_n) = \left(\frac{\exp(y_{ni})}{\sum_i \exp(y_{ni})}\right)_{i=1}^k$ is the softmax function. This framework is illustrated in Fig. 3.1.

The observed class and cluster labels are represented as $\boldsymbol{W} = \{\{\boldsymbol{w}_{nl}^{(1)}\}, \{\boldsymbol{w}_{nm}^{(2)}\}\}$ where $\boldsymbol{w}_{nl}^{(1)}$ is the 1-of-k representation of class label of the n^{th} object given by the l^{th} classifier, and $\boldsymbol{w}_{nm}^{(2)}$ is the 1-of-$k^{(m)}$ representation of cluster label assigned to the n^{th} object by the m^{th} clusterer. A generative model is proposed to explain the observations $\boldsymbol{W}$, where each object $\mathbf{x}_n$ has an underlying mixed-membership to the k different classes. Let $f(\boldsymbol{y}_n)$ denote the latent mixed-membership vector for $\mathbf{x}_n$. $\boldsymbol{y}_n$ is sampled from a normal distribution. Also, corresponding to the i^{th} class and m^{th} base clustering, we assume a multinomial distribution $\boldsymbol{\beta}_{mi}$ over the cluster labels of the m^{th} base clustering. Therefore, $\boldsymbol{\beta}_{mi}$ is of dimension $k^{(m)}$ and $\sum_{j=1}^{k^{(m)}} \beta_{mij} = 1$ if the m^{th} base clustering has $k^{(m)}$ clusters. The data generative process, whose corresponding graphical model is shown in Fig. 3.2, can be summarized as follows:

- For each $\mathbf{x}_n \in \mathcal{X}$, choose $\boldsymbol{y}_n \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu} \in \mathbb{R}^k$ is the mean and $\boldsymbol{\Sigma} \in \mathbb{R}^{k \times k}$ is the covariance.
- Choose $\boldsymbol{\theta}_n \sim \mathcal{N}(\boldsymbol{y}_n, \delta^2 I_k)$, where $\delta^2 \geq 0$ is the scaling

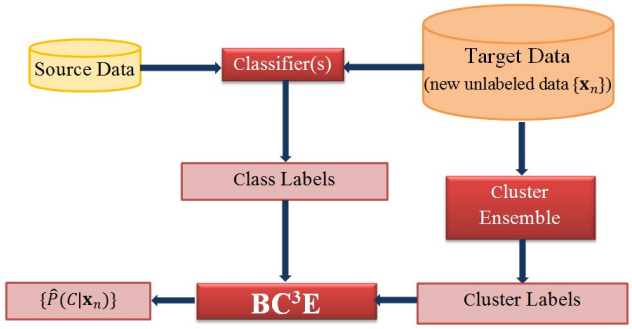

Figure 3.1: Combining Classifiers and Clusterers.

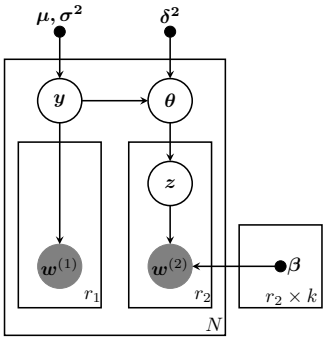

Figure 3.2: Graphical Model for **BC^{3}E**

factor of the covariance of the normal distribution centered at $\boldsymbol{y}_n$, and I_k is the identity $k \times k$ matrix.

- $\forall l \in \{1, 2, \cdots, r_1\}$, choose $\boldsymbol{w}_{nl}^{(1)} \sim f(\boldsymbol{y}_n)$.
- $\forall m \in \{1, 2, \cdots, r_2\}$:
  1. Choose $\boldsymbol{z}_{nm} \sim f(\boldsymbol{\theta}_n)$, where $\boldsymbol{z}_{nm}$ is a k-dimensional vector with 1-of-k representation.
  2. Choose $\boldsymbol{w}_{nm}^{(2)} \sim \text{multinomial}(\boldsymbol{\beta}_{r\boldsymbol{z}_{nm}})$.

The observed class labels $\{\boldsymbol{w}_{nl}^{(1)}\}$ are assumed to be sampled from the latent mixed-membership vector $f(\boldsymbol{y}_n)$. If the n^{th} object is sampled from the i^{th} class in the m^{th} base clustering (implying $z_{nmi} = 1$), then its cluster label will be sampled from the multinomial distribution $\boldsymbol{\beta}_{mi}$. This particular generative process is analogous to the one used by the Bayesian Cluster Ensemble in [21]. The fact that $\boldsymbol{\theta}_n$ is sampled from $\mathcal{N}(\boldsymbol{y}_n, \delta^2 I_k)$ needs further clarification. In practice, the observed class labels and cluster labels carry different intrinsic weights. If the observations from the classifiers are assigned too much weight compared to those from clustering, there is little hope for the clustering to enhance classification. Similarly, if the observations from the clustering are given too much of importance, the classification performance might deteriorate. Ideally, the unsupervised information is only expected to enhance the classification accuracy.

Aimed at building a "safe" model that can intelligently utilize or reject the unsupervised information, $\boldsymbol{\theta}_n$ is sampled from $\mathcal{N}(\boldsymbol{y}_n, \delta^2 I_k)$ where the parameter δ decides how much the observations from the clusterings can be trusted. If δ^2 is a large positive number, $\boldsymbol{y}_n$ does not have to explain the posterior of $\boldsymbol{\theta}_n$. From the generative model perspective, this means that the sampled value of $\boldsymbol{\theta}_n$ is not governed by $\boldsymbol{y}_n$ anymore as the distribution has very large variance. On the other hand, if δ^2 is a small positive number, $\boldsymbol{y}_n$ has to explain the posterior of $\boldsymbol{\theta}_n$ and hence the observations from the clustering. Therefore, the posteriors of $\{\boldsymbol{y}_n\}$ are expected to get more accurate compared to the case if they only had to explain the classification results. A concrete quantitative argument for this intuitive statement will be presented later. Note that sampling both $\boldsymbol{y}_n$ and $\boldsymbol{\theta}_n$ from normal distributions allow the mathematical formulation to be simpler and intuitive. A Dirichlet-multinomial model, instead, would have complicated the inference process.

To address the log-likelihood function of **BC^{3}E**, let us denote the set of hidden variables by $\boldsymbol{Z} = \{\{\boldsymbol{y}_n, \{\boldsymbol{\theta}_n\}, \{\boldsymbol{z}_{nm}\}\}$. The model parameters can conveniently be represented by $\boldsymbol{\zeta}_0 = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}, \delta^2, \{\boldsymbol{\beta}_{mi}\}\}$. The joint distribution of the hidden and observed variables is given in Eq. (1). The inference and estimation is performed using Variational Expectation-Maximization (**VEM**) to avoid computational intractability due to the coupling between $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$.

3.1 Approximate Inference and Estimation:

3.1.1 Inference: To obtain a tractable lower bound on the observed log-likelihood, we specify a fully factorized distribution to approximate the true posterior of the hidden variables, as given in Eq. (2). Here $\boldsymbol{y}_n \sim \mathcal{N}(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$, $\boldsymbol{\theta}_n \sim \mathcal{N}(\boldsymbol{\epsilon}_n, \boldsymbol{\Delta}_n)$ $\forall n \in \{1, 2, \cdots, N\}$, $\boldsymbol{z}_{nm} \sim \text{multinomial}(\boldsymbol{\phi}_{nm})$ $\forall n \in \{1, 2, \cdots, N\}$ and $\forall m \in \{1, 2, \cdots, r_2\}$, and $\boldsymbol{\zeta}_n = \{\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n, \boldsymbol{\epsilon}_n, \boldsymbol{\Delta}_n), \{\boldsymbol{\phi}_{nm}\}\}$ – the set of variational parameters corresponding to the n^{th} object. Further, $\boldsymbol{\mu}_n, \boldsymbol{\epsilon}_n \in \mathbb{R}^k$, $\boldsymbol{\Sigma}_n, \boldsymbol{\Delta}_n \in \mathbb{R}^{k \times k}$ $\forall n$ and $\boldsymbol{\phi}_{nm} = (\phi_{nmi})_{i=1}^k$ $\forall n, m$. To work with less parameters, all the covariance matrices are assumed to be diagonal. Therefore, $\boldsymbol{\Sigma} = \text{diag}\left((\sigma_i)_{i=1}^k\right)$, $\boldsymbol{\Sigma}_n = \text{diag}\left((\sigma_{ni})_{i=1}^k\right)$, and $\boldsymbol{\Delta}_n = \text{diag}\left((\delta_{ni})_{i=1}^k\right)$. Using Jensen's inequality, a lower bound on the observed log-likelihood can be derived as:

$$\begin{aligned} \log[p(\boldsymbol{X}|\boldsymbol{\zeta}_0)] &\geq \mathbf{E}_{q(\boldsymbol{Z})}\left[\log[p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\zeta}_0)]\right] + H(q(\boldsymbol{Z})) \\ &= \mathcal{L}(q(\boldsymbol{Z})) \end{aligned} \tag{3.1}$$

Joint Distribution of $\mathbf{BC^3E}$
$p(\mathbf{X}, \boldsymbol{Z}\|\boldsymbol{\zeta}_0) = \prod_{n=1}^{N} p(\boldsymbol{y}_n\|\boldsymbol{\mu}, \boldsymbol{\Sigma}) p(\boldsymbol{\theta}_n\|\boldsymbol{y}_n, \delta^2 I_k) \prod_{l=1}^{r_1} p(w_{1nl}\|f(\boldsymbol{y}_n)) \prod_{m=1}^{r_2} p(\boldsymbol{z}_{nm}\|f(\boldsymbol{\theta}_n)) p(w_{2nm}\|\boldsymbol{\beta}, \boldsymbol{z}_{nm})$.(1)
Variational Distribution of $\mathbf{BC^3E}$
$q(\boldsymbol{Z}\|\{\boldsymbol{\zeta}_n\}_{n=1}^{N}) = \prod_{n=1}^{N} q(\boldsymbol{y}_n\|\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n) q(\boldsymbol{\theta}_n\|\boldsymbol{\epsilon}_n, \boldsymbol{\Delta}_n) \prod_{m=1}^{r_2} q(\boldsymbol{z}_{nm}\|\boldsymbol{\phi}_{nm})$.(2)

Table 3.1: Distributions in $\mathbf{BC^3E}$

where $H(q(\boldsymbol{Z})) = -\mathbf{E}_{q(\boldsymbol{Z})}[\log[q(\boldsymbol{Z})]]$ is the entropy of the variational distribution $q(\boldsymbol{Z})$, and $\mathbf{E}_{q(\boldsymbol{Z})}[.]$ is the expectation w.r.t $q(\boldsymbol{Z})$. Let $\mathcal{Q}$ be the set of all distributions having a fully factorized form as given in Eq. (2). The optimal distribution that produces the tightest possible lower bound $\mathcal{L}$ is given by:

$$q^* = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}(p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\zeta}_0)||q(\boldsymbol{Z})). \tag{3.2}$$

In equations (3), (5), (7), (9), (11), (12) and (13) in Table 3.1, the optimal values of the variational parameters that satisfy Eq. (3.2) are presented. Since the logistic normal distribution is not conjugate to multinomial, the update equations of all the parameters cannot be obtained in closed form. For the parameters that do not have a closed form solution for the update, we just present the part of the objective function that depends on the concerned parameter and some numeric optimization method has to be used for optimizing the lower bound. Since $\boldsymbol{\phi}_{nm}$ is a multinomial distribution, the updated values of the k components should be normalized to unity. Note that the optimal value of one of the variational parameters depends on the others and, therefore, an iterative optimization is adopted to minimize the lower bound till convergence is achieved.

Equations (5) and (7) present updates for two new parameters. These parameters come from $\mathbb{E}_q(\log p(\boldsymbol{w}_{nl}^{(1)}|f(\boldsymbol{y}_n)))$ and $\mathbb{E}_q(\log p(\boldsymbol{z}_{nm}|f(\boldsymbol{\theta}_n)))$ respectively. Both of these integrations do not have analytic solution and hence a first order Taylor approximation is utilized as also done in [5]. A closer inspection of (11) reveals that δ^2 appears in the denominator of the term $\sum_{i=1}^{k}(\mu_{ni} - \epsilon_{ni})^2/\delta^2$ in the objective. Hence, larger values of δ^2 will nullify any effect from $\boldsymbol{\epsilon}_n$ which, in turn, is affected by the observations $\{\boldsymbol{w}_{nm}^{(2)}\}$ (as is obvious from (13)). On the other hand, if δ^2 is small enough, $\boldsymbol{\epsilon}_n$ can strongly impact the values of $\boldsymbol{\mu}_n$.

3.1.2 Estimation: For estimation, we maximize the optimized lower bound in Eq. (3.1) w.r.t the free model parameters $\boldsymbol{\zeta}_0$. The optimal values of the model parameters are presented in equations (4), (6) and (8). Since $\boldsymbol{\beta}_{mi}$ is a multinomial distribution, the updated values of $k^{(m)}$ components should be normalized to unity. However, no closed form of update exists for $\boldsymbol{\sigma}^2$, and a numeric optimization method has to be resorted to. The part of the objective function that depends on $\boldsymbol{\sigma}^2$ is provided in Eq. (10). Once the optimization in M-step is done, E-step starts and the iterative update is continued till convergence. The variational parameters $\{\boldsymbol{\mu}_n\}_{n=1}^{N}$ are then investigated which serve as proxy for the refined posterior estimates of $\{\boldsymbol{y}_n\}_{n=1}^{N}$. The main steps of inference and estimation are concisely presented in Algorithm 1.

Algorithm 1 Learning $\mathbf{BC^3E}$

Input: $\boldsymbol{W}$.
Output: $\boldsymbol{\theta}^m, \{\boldsymbol{\mu}_n\}_{n=1}^{N}$.

Initialize $\boldsymbol{\theta}^m$, $\{\boldsymbol{\zeta}_n\}_{n=1}^{N}$.
Until Convergence
E-Step
Until Convergence
1. Update ϕ_{nmi} using Eq. (3) $\forall n, m, i$. Normalize $\boldsymbol{\phi}_{nm}$.
2. Update κ_n using Eq. (5) $\forall n \in \{1, 2, \cdots, N\}$.
3. Update ξ_n using Eq. (7) $\forall n \in \{1, 2, \cdots, N\}$.
4. Maximize (9) w.r.t. $\boldsymbol{\delta}_n^2$ $\forall n$ s.t. $\boldsymbol{\delta}_n^2 \geq 0$.
5. Maximize (11) w.r.t. $\boldsymbol{\mu}_n$ $\forall n$.
6. Maximize (12) w.r.t. $\boldsymbol{\sigma}_n^2$ $\forall n$ s.t. $\boldsymbol{\sigma}_n^2 \geq \mathbf{0}$.
7. Maximize (13) w.r.t. $\boldsymbol{\epsilon}_n$ $\forall n$.
M-Step
8. Update $\boldsymbol{\mu}$ using Eq. (4).
9. Update β_{mij} using Eq. (6) $\forall m, i, j$. Normalize $\boldsymbol{\theta}_{mi}$.
10. Update δ^2 using Eq. (8).
11. Maximize (10) w.r.t. $\boldsymbol{\sigma}^2$ s.t. $\boldsymbol{\sigma}^2 \geq \mathbf{0}$.

4 Privacy Preserving Learning

Most of the privacy-aware distributed data mining techniques developed so far have focused on classification or on association rules [4, 11]. There has also been some work on distributed clustering for *vertically partitioned data* (different sites contain different attributes/features of a common set of records/objects) [15], and on parallelizing clustering algorithms for *horizontally partitioned data* (i.e. the objects are distributed amongst the sites, which record the same set of features for each object) [9]. These techniques, however, do not specifically address privacy issues, other than through encryption [20].

This is also true of earlier, data-parallel methods [9] that are susceptible to privacy breaches, and also

Update Equations	
$\phi^*_{nmi} \propto \exp\left(\epsilon_{ni} + \sum_{j=1}^{k^{(m)}} \beta_{mij} w^{(2)}_{nmj}\right) \ \forall n, m, i.$ (3)	$\boldsymbol{\mu}^* = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{\mu}_n.$ (4)
$\kappa^*_n = \sum_{i=1}^{k} \exp(\mu_{ni} + \sigma^2_{ni}/2) \ \forall n.$ (5)	$\beta^*_{mij} \propto \sum_{n=1}^{N} \phi_{nmi} w^{(2)}_{nmj} \ \forall j \in 1, 2, \cdots, k^m.$ (6)
$\xi^*_n = \sum_{i=1}^{k} \exp(\epsilon_{ni} + \delta^2_{ni}/2) \ \forall n.$ (7)	$\delta^2 = \frac{1}{Nk}\sum_{n=1}^{N}\sum_{i=1}^{k}\left[(\epsilon_{ni} - \mu_{ni})^2 + \sigma^2_{ni} + \delta^2_{ni}\right].$ (8)
$\mathcal{L}_{[\boldsymbol{\delta}^2_n]} = -\frac{1}{2}\sum_{i=1}^{k}\frac{\delta^2_{ni}}{\delta^2} - \frac{1}{2}\sum_{i=1}^{k}\log(\delta^2_{ni}) - \frac{r_2}{\xi_n}\sum_{i=1}^{k}\exp(\epsilon_{ni} + \delta^2_{ni}/2).$ (9)	$\mathcal{L}_{[\boldsymbol{\sigma}^2]} = -\frac{N}{2}\sum_{i=1}^{k}\log(\sigma^2_i) - \frac{1}{2}\sum_{n=1}^{N}\sum_{i=1}^{k}[\frac{\sigma^2_{ni} + (\mu_{ni} - \mu_i)^2}{\sigma^2_i}].$ (10)
$\mathcal{L}_{[\boldsymbol{\mu}_n]} = -\frac{1}{2}\sum_{i=1}^{k}\frac{(\mu_{ni} - \mu_i)^2}{\sigma^2_i} - \frac{1}{2\delta^2}\sum_{i=1}^{k}(\mu_{ni} - \epsilon_{ni})^2 + \sum_{l=1}^{r_1}\sum_{i=1}^{k} w^{(1)}_{nli}\mu_{ni} - \frac{r_1}{\xi_n}\sum_{i=1}^{k}\exp(\mu_{ni} + \sigma^2_{ni}/2).$ (11)	
$\mathcal{L}_{[\boldsymbol{\sigma}^2_n]} = -\frac{1}{2}\sum_{i=1}^{k}\frac{\sigma^2_{ni}}{\sigma^2_i} - \frac{1}{2}\sum_{i=1}^{k}\log(\sigma^2_{ni}) - \frac{1}{2}\sum_{i=1}^{k}\frac{\sigma^2_{ni}}{\delta^2} - \frac{r_1}{\kappa_n}\sum_{i=1}^{k}\exp(\mu_{ni} + \sigma^2_{ni}/2).$ (12)	
$\mathcal{L}_{[\boldsymbol{\epsilon}_n]} = \sum_{m=1}^{r_2}\sum_{i=1}^{k}\phi_{nmi}\epsilon_{ni} - \frac{1}{\xi_n}\sum_{i=1}^{k}\exp(\epsilon_{ni} + \delta^2_{ni}/2) - \frac{1}{2}\sum_{i=1}^{k}\frac{(\epsilon_{ni} - \mu_{ni})^2}{\delta^2}.$ (13)	

Table 3.2: Equations for update of variational and model parameters in **BC^{3}E**

need a central planner that dictates what algorithm runs on each site. Finally, recent works on distributed differential privacy focus on query processing rather than data mining [7].

In the sequel, we show that the inference and estimation in **BC^{3}E** using **VEM** allows solving the cluster ensemble problem in a way that preserves privacy. Depending on how the objects with their cluster/class labels are distributed in different "data sites", we can have the following three scenarios.

4.1 Row Distributed Ensemble: In the row distributed framework, the test set $\mathcal{X}$ is partitioned into D parts and different parts are assumed to be at different locations. The objects from partition d are denoted by $\mathcal{X}_d$ so that $\mathcal{X} = \cup_{d=1}^{D}\mathcal{X}_d$. Now, a careful look at the E-step equations reveal that the update of variational parameters corresponding to each object in a given iteration is independent of those of other objects. Therefore, we can maintain a client-server based framework where the server only updates the model parameters and the clients (there should be as many number of clients as there are distributed data sites) update the variational parameters.

For instance, consider a situation where a dataset is partitioned into two subsets $\mathcal{X}_1$ and $\mathcal{X}_2$ and these two subsets are located in two different data sites. Data site 1 has access to $\mathcal{X}_1$ and a set of clustering and classification results pertaining to objects belonging to $\mathcal{X}_1$. Similarly, data site 2 has access to $\mathcal{X}_2$ and a set of clustering and classification results corresponding to $\mathcal{X}_2$. Further assume that a set of distributed classification (clustering) algorithms were used to generate the class (cluster) labels of the objects belonging to each set. Now, data site 1 can update the variational parameters $\boldsymbol{\zeta}_n$, $\forall \mathbf{x}_n \in \mathcal{X}_1$. Similarly, data site 2 can update the variational parameters for all objects $\mathbf{x}_n \in \mathcal{X}_2$. Once the variational parameters are updated in the E-step, the server gathers information from two sites and updates the model parameters. Now, a closer inspection of the M-step update equations reveals that each of them contains a summation over the objects. Therefore, individual data sites can send only some collective information to the server without transgressing privacy. For example, consider the update equation for β_{mij}. Eq. (6) can be broken as follows:

$$\beta_{mij}{}^* \propto \sum_{x_n \in \mathcal{X}_1} \phi_{nmj} w^{(2)}_{nmj} + \sum_{x_n \in \mathcal{X}_2} \phi_{nmj} w^{(2)}_{nmj} \quad (4.3)$$

The first and second terms can be calculated in data sites 1 and 2 separately and sent to the server where the two terms can be added and β_{mij} can get updated $\forall m, i, j$. Similarly, the other M-step update equations (performed by the server in an analogous way) also do not reveal any information about class or cluster labels of objects belonging to different data sites.

4.2 Column Distributed Ensemble: In the column distributed framework, different data sites share the same set of objects but only a subset of base clusterings or classification results are available to each data

site. For example, consider that we have two data sites and four sets of class and cluster labels and each data site has access to only two sets of classification or clustering results. Assume that data site 1 has access to the 1st and 2nd classification and clustering results and data site 2 has access to the rest of the results. As in the earlier case, a single server and two clients (corresponding to two different data sites) are maintained. Since each data site has access to all the objects, it is necessary to share the variational parameters corresponding to these objects. Therefore, $\{\kappa_n, \xi_n, \boldsymbol{\mu}_n, \boldsymbol{\sigma}_n, \boldsymbol{\epsilon}_n, \boldsymbol{\delta}_n\}_{n=1}^N$ are all updated in the server.

The site (and object) specific variational parameters $\{\phi_{nmi}\}$, however, cannot be shared and should be updated in individual sites. This means that the updates (5), (7), (11), (13), (9) and (12) should be performed in the server. On the other hand, the update for $\{\phi_{nmi}\}\forall n, i$ and $m \in \{1, 2\}$ (corresponding to the 1nd and 2nd clustering or classification results) should be performed in data site 1. Similarly, the update for $\{\phi_{nmi}\}\ \forall n, i$ and $m \in \{3, 4\}$ has to be performed in data site 2. However, while updating $\{\boldsymbol{\mu}_n\}$, the calculation of the term $\sum_{l=1}^{r_1}\sum_{i=1}^{k} w_{nli}^{(1)}\mu_{ni}$ has to be performed without revealing the class labels $\{\boldsymbol{w}_{nl}^{(1)}\}$ to the server. To that end, it can be rewritten as:

(4.4)
$$\sum_{l=1}^{r_1}\sum_{i=1}^{k} w_{nli}^{(1)}\mu_{ni} = \sum_{l=1}^{2}\sum_{i=1}^{k} w_{nli}^{(1)}\mu_{ni} + \sum_{l=3}^{4}\sum_{i=1}^{k} w_{nli}^{(1)}\mu_{ni},$$

where the first term can be computed in data site 1 and the second term can be computed by data site 2 and then can be added in the server. It can be seen that $\{\boldsymbol{w}_{nl}^{(1)}\}$ can never be recovered by the server and hence privacy is ensured in the updates of the E-step. Except for $\{\beta_{mij}\}$, all other model parameters can be updated in the server in the M-step. However, the parameters $\{\beta_{mij}\}$ have to be updated separately inside the clients. Since $\{\beta_{mij}\}$ do not appear in any update equation performed in the server, there is no need to send these parameters to the server either. Therefore, in essence, the clients update the parameters $\{\phi_{nmi}\}$ and $\{\beta_{mij}\}$ in E-step and M-step respectively, and the server updates the remaining parameters.

4.3 Arbitrarily Distributed Ensemble: In an arbitrarily distributed ensemble, each data site has access to only a subset of the data points or a subset of the classification and clustering results. Fig. 4.1 shows a situation with arbitrarily distributed ensemble with six data sites. We now refer to Fig. 4.2 and explain the privacy preserved EM update for this setting. As before, corresponding to each different data site, a client node is created. Clients that share a subset of the objects should have access to the variational parameters corresponding to common objects. To highlight the sharing of objects by clients, the test set $\mathcal{X}$ is partitioned into four subsets — $\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3$ and$\mathcal{X}_4$ as shown in Fig. 4.1. Similarly, the columns are also partitioned into three subsets: G_1, G_2, and G_3.

Now, corresponding to each row partition, an "Auxiliary Server"(AS) node is created. Each AS updates the variational parameters corresponding to a set of shared objects. For example, in Fig. 4.3, AS_1 updates the variational parameters corresponding to $\mathcal{X}_1$ (using equations (7), (5), (11), (12), (13), and (9)). However, any variational parameter that is specific to both an object and a column is updated separately inside the corresponding client (and hence it is connected with C_1 and C_2). Therefore, $\{\phi_{nmi} : n \in \mathcal{X}_1, m \in G_1\}$ are updated inside client 1 and $\{\phi_{nmi} : n \in \mathcal{X}_1, m \in G_2 \cup G_3\}$ are updated inside client 2 (using Eq. (3)). Once all variational parameters are updated in the E-step, M-step starts. Corresponding to each column partition, an "Auxiliary Client" (AC) node is created. This node updates the model parameters β_{mij} (using Eq. (6)) which are specific to columns belonging to G_1. Since C_1, C_3, and C_5 share the columns from the subset G_1, AC_1 is connected with these three nodes in Fig. 4.3. The remaining model parameters are, however, updated in a "Server" (using equations (4), (8), (10)).

In Fig. 4.3, the bidirectional edges indicate that messages are sent to and from the connecting nodes. We have avoided separate arrows for each direction only to keep the figure uncluttered. The edges are also numbered near to their origin. For a comprehensive understanding of the privacy preservation, the messages transfered through each edge have also been enlisted in the supplementary material. The messages sent from the auxiliary servers to the main server are of the form given in Eq. (4.3) and are denoted as "partial sums". Expectedly, messages sent out from a client node are "masked" in such a way that no other node can decode the cluster labels or class labels of points belonging to that client. This approach is completely general and will work for any arbitrarily partitioned ensemble given that each partition contains at least two sets of classification results. Note that the ACs and ASs are only helpful in conceptual understanding of the parameter update and sharing. In practice, there is no real need for these extra storage devices/locations. Client nodes can themselves take the place of ASs and ACs and even the main server as long as the updates are performed in

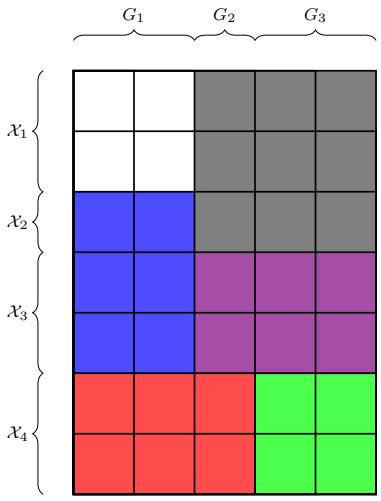

Figure 4.1: Arbitrarily Distributed Ensemble

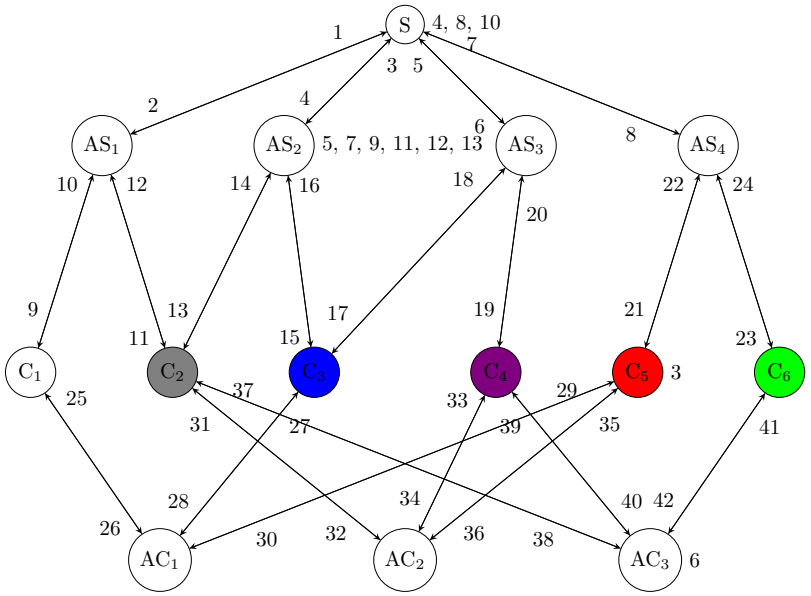

Figure 4.2: Parameter Update for Arbitrarily Distributed Ensemble

proper sequence[1].

5 Experiments

In this section, two different sets of experiments are reported. The first set is for transfer learning with a text classification data from eBay Inc. The other set is for non-transductive semisupervised learning where some publicly available datasets are used to simulate the working environment of **BC^{3}E**.

5.1 Transfer Learning: To show the capability of **BC^{3}E** in solving transfer learning problems, we use a large scale text classification dataset from eBay Inc. The training data consists of 83 million items sold over a three month period of time and the test set contains several millions of items sold a few days after the training period. More details about the dataset can be found in [18]. eBay organizes items into a six-level category structure where there are 39 top level nodes called *meta categories* and 20K+ bottom level nodes called *leaf categories*. The dataset is generated when users provide the titles of items they intend to sell on eBay. Each title is limited to 50 characters, based on which the user gets recommendation of some *leaf categories* the item should belong to. Such categorization of the item helps a seller list an item in the correct branch of the product list, thereby allowing a buyer more easily search through a list of few million items sold via eBay every single day. A carefully designed k-Nearest Neighbor (k-NN) classifier (with the help of improved search engine algorithms) categorizes each of the items in less than 100 ms [18]. However, due to the large number of categories (20K), items belonging to similar types of categories often get misclassified.

To avoid such confusion, larger categories are formed by aggregating examples from categories which are relatively difficult to separate. Such aggregation is easy once the confusion matrix of the classification, obtained from a development dataset, is partitioned and strongly connected vertices (each vertex representing one of 20K *leaf categories*) are identified from the confusion graph, thereby forming a set of cliques which represent the large categories. Note that the large categories so discovered might not at all follow the internal hierarchy that is maintained. Next, clustering is performed with examples belonging to each of the large categories and the clustering results, along with the predictions from k-NN classification, are fed to **BC^{3}E** (and also to its competitors *i.e.* **C^{3}E**, **BGCM**, and **LWE**). The idea here is to first reduce the classification space and then use unsupervised information to refine the predictions from k-NN on a smaller number of categories. The number of *leaf categories* belonging to such large categories usually varies between 4-10.

However, the dataset is very dynamic and, typically over a span of three months, 20% of new words are added to the existing vocabulary. One can retrain the existing k-NN classifier every three months, but

[1]Note that such framework allows running the updates of the same stage in parallel in different sites, thereby saving the computation time in large scale implementations.

the training process requires collecting new labeled data which is time consuming and expensive. One can additionally design classifiers to segregate examples belonging to each of the large categories. However, such approach might not improve much upon the performance of the initial k-NN classifier if the data changes so frequently. Therefore, we require a system that can adaptively predict newer examples without retraining the existing classifier or employing another set of classification algorithms. **BC^{3}E** is useful in such settings. The parameter δ can adjust the weights of prediction from classifiers and unsupervised information. As the results reported in Table 5.1 reveal, as long as the classification performance is not that poor, **BC^{3}E** can improve on the performance of k-NN using the clustering ensemble.

The column "Group ID" denotes anonymized groups representing different large categories. $|\mathcal{X}|$ shows the number of examples in the test data. The column "C^{3}E-Ideal" shows the performance of **C^{3}E** if the correct tuning parameter for **C^{3}E** were known. For a transfer learning problem, estimating such tuning parameter requires some labeled data from the target set which is not available in our setting. If the tuning parameter is chosen from cross-validation on the training data, the final prediction on target set can get affected adversely if the underlying distribution changes (and in fact it does in our experiments). Therefore, we need to adopt a fail-safe approach where we can do at least as good as the k-NN prediction. The results reveal that **BC^{3}E** significantly outperforms **BGCM** and **LWE**, and sometimes achieves as good a performance as **C^{3}E**-Ideal (*i.e.* when correct tuning parameter of **C^{3}E** is known). The performance of **C^{3}E**-Ideal can essentially be considered as the best accuracy one could achieve from the given inputs (*i.e.* class and cluster labels) using other existing algorithms — **BGCM**, **LWE**, **C^{3}E** — that work on the same design space. Though **BGCM** has a tuning parameter, its variation did not affect performance much and we just report results corresponding to unity value of this parameter.

5.2 Semi-supervised Learning: Six datasets are used in our experiments for semi-supervised learning: *Half-Moon* (a synthetic dataset with two half circles representing two classes), *Circles* (another synthetic dataset that has two-dimensional instances that form two concentric circles — one for each class), and four datasets from the *Library for Support Vector Machines* — *Pima Indians Diabetes*, *Heart*, *German Numer*, and *Wine*. In order to simulate semi-supervised settings where there is a very limited amount of labeled instances, small percentages (see the values reported in Table 5.2) of the instances are randomly selected for training, whereas the remaining instances are used for testing. We perform 20 trials for every dataset. For experiments with **BGCM**, and **C^{3}E**, the parameters reported in [13] and [2] are used respectively. The parameters of **BC^{3}E** are initialized randomly and approximately 10 EM iterations are enough to get the results reported in Table 5.2. The classifier ensemble consists of decision tree (C4.5), linear discriminant, and generalized logistic regression. Cluster ensembles are generated from multiple runs of k-means [2]. **LWE** [12] is better suited for transfer learning applications and hence has been left out from comparison. The column "Best" in Table 5.2 refers to the performance of the best classifier in the ensemble. Note that **BC^{3}E** has superior performance for the most difficult problems, where one has an incentive to use a more complex mechanism. Most importantly, **BC^{3}E** has the privacy preserving property not present in any of its counterparts.

6 Conclusion and Future Work

The **BC^{3}E** model proposed in this paper has been shown to be useful for difficult non-transductive semisupervised and transfer learning problems. A good tradeoff between accuracy and privacy has also been established empirically – a property absent in any of **BC^{3}E**'s competitors. With minor modification, **BC^{3}E** can also handle soft outputs from classification and clustering ensembles which can further improve the results.

Acknowledgement

This work has been supported by NSF Grants (IIS-0713142 and IIS-1016614), ONR Grant (ATL N00014-11-1-0105) and by the Brazilian Research Agencies FAPESP and CNPq.

References

[1] A. Acharya, E.R. Hruschka, and J. Ghosh, *A privacy-aware bayesian approach for combining classifier and cluster ensembles*, in SocialCom/PASSAT, 2011, pp. 1169–1172.

[2] A. Acharya, E.R. Hruschka, J. Ghosh, and S. Acharyya, *An optimization framework for semi-supervised and transfer learning using multiple classifiers and clusterers*, CoRR, abs/1206.0994 (2012).

[3] A. Acharya, E. R. Hruschka, J. Ghosh, and S. Acharyya, *C^{3}E: A Framework for Combining Ensembles of Classifiers and Clusterers*, in 10th Int. Workshop on MCS, 2011.

[4] D. Agrawal and C. C. Aggarwal, *On the design and quantification of privacy preserving data mining algorithms*, in Symposium on Principles of Database Systems, 2001.

Group ID	$\lvert\mathcal{X}\rvert$	k-NN	BGCM	LWE	C^{3}E-Ideal	BC^{3}E
42	1299	64.90	73.78 (± 0.94)	76.86 (± 1.01)	83.99 (± 0.41)	83.68 (± 1.09)
84	611	63.67	69.23 (± 0.17)	75.24 (± 0.26)	81.18 (± 0.16)	76.27 (± 1.31)
86	2381	77.66	84.33 (± 2.74)	83.29 (± 1.02)	92.78 (± 0.35)	87.20 (± 0.91)
67	789	72.75	72.75 (± 0.07)	78.03 (± 0.72)	82.64 (± 0.82)	81.75 (± 1.37)
52	1076	76.95	77.01 (± 1.18)	77.49 (± 1.41)	88.38 (± 0.22)	85.04 (± 2.14)
99	827	84.04	85.12 (± 0.52)	86.90 (± 0.92)	91.54 (± 0.27)	91.17 (± 0.82)
48	3445	86.33	86.19 (± 0.25)	90.38 (± 1.03)	92.71 (± 0.31)	92.71 (± 1.16)
94	440	79.32	81.08 (± 0.73)	82.52 (± 0.83)	85.45 (± 0.09)	85.45 (± 0.79)
35	4907	82.41	82.10 (± 0.37)	85.08 (± 1.39)	88.16 (± 0.17)	88.22 (± 1.21)
45	1952	74.80	73.12 (± 0.81)	73.64 (± 1.68)	84.32 (± 0.23)	77.97 (± 0.47)

Table 5.1: Performance of **BC^{3}E** on text classification data — Avg. Accuracies ±(Standard Deviations).

Dataset (% of tr. data)	$\lvert\mathcal{X}\rvert$	Ensemble	Best	BGCM	C^{3}E	BC^{3}E
Half-moon(2%)	784	92.53(±1.83)	93.02(±0.82)	92.16(±1.47)	**99.64**(±0.08)	98.23(±2.03)
Circles(2%)	1568	60.03(±8.44)	95.74(±5.15)	78.67(±0.54)	**99.61**(±0.83)	97.91(±0.74)
Pima(2%)	745	68.16(±5.05)	69.93(±3.68)	69.21(±4.83)	70.31(±4.44)	**72.83**(±0.49)
Heart(7%)	251	77.77(±2.55)	79.22(±2.20)	82.78(±4.82)	**82.85**(±5.25)	82.53(±1.14)
G. Numer(10%)	900	70.96(±1.00)	70.19(±1.52)	73.70(±1.06)	74.44(±3.44)	**74.61**(±1.62)
Wine(10%)	900	79.87(±5.68)	80.37(±5.47)	75.37(±13.66)	**83.62**(±6.27)	82.20(±1.07)

Table 5.2: Comparison of **BC^{3}E** with **C^{3}E** and **BGCM** — Avg. Accuracies ±(Standard Deviations).

[5] D.M. Blei and J.D. Lafferty, *A correlated topic model of science*, Annals of Applied Statistics, 1 (2007), pp. 17–35.

[6] A. Blum, *On-line algorithms in machine learning*, in Online Algorithms: The State of the Art, Fiat and Woeginger, eds., LNCS Vol.1442, Springer, 1998.

[7] R. Chen, A. Reznichenko, P. Francis, and J. Gehrke, *Towards statistical queries over distributed private user data*, in Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation, NSDI'12, 2012.

[8] H. A. Chipman, E. I. George, and R. E. McCulloch, *Bayesian ensemble learning*, in Proc. of Neural Information Processing Systems, 2006, pp. 265–272.

[9] I. S. Dhillon and D. S. Modha, *A data-clustering algorithm on distributed memory multiprocessors*, in Proc. Large-scale Parallel Knowledge Discovery and Data Mining Systems Workshop, ACM SIGKnowledge Discovery and Data Mining, August 1999.

[10] N. U. Edakunni and S. Vijayakumar, *Efficient online classification using an ensemble of bayesian linear logistic regressors*, in 8th Int. Workshop on MCS, 2009, pp. 102–111.

[11] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, *Privacy preserving mining of association rules*, in Knowledge Discovery and Data Mining, 2002.

[12] J. Gao, W. Fan, J. Jiang, and J. Han, *Knowledge transfer via multiple model local structure mapping*, in Proc. of KDD, 2008, pp. 283–291.

[13] J. Gao, F. Liang, W. Fan, Y. Sun, and J. Han, *Graph-based consensus maximization among multiple supervised and unsupervised models*, in Proc. of NIPS, 2009, pp. 1–9.

[14] Z. Ghahramani and H. Kim, *Bayesian classifier combination*, tech. report, 2003.

[15] E. Johnson and H. Kargupta, *Collective, hierarchical clustering from distributed, heterogeneous data*, in Large-Scale Parallel Knowledge Discovery and Data Mining Systems, vol. 1759 of LNCScience, 1999, pp. 221–244.

[16] S. Merugu and J. Ghosh, *Privacy perserving distributed clustering using generative models*, in Proc. of ICDM, Nov, 2003, pp. 211–218.

[17] N. C. Oza and K. Tumer, *Classifier ensembles: Select real-world applications*, Inf. Fusion, 9 (2008), pp. 4–20.

[18] D. Shen, J. Ruvini, and B. Sarwar, *Large-scale item categorization for e-commerce*, in Proc. of CIKM, New York, NY, USA, 2012, ACM, pp. 595–604.

[19] D. L. Silver and K. P. Bennett, *Guest editor's introduction: special issue on inductive transfer learning*, Machine Learning, 73 (2008), pp. 215–220.

[20] J. Vaidya and C. Clifton, *Privacy-preserving k-means clustering over vertically patitioned data*, in KDD, 2003, pp. 206–215.

[21] H. Wang, H. Shan, and A. Banerjee, *Bayesian cluster ensembles*, Statistical Analysis and Data Mining, 1 (2011), pp. 1–17.

Using Both Latent and Supervised Shared Topics for Multitask Learning

Ayan Acharya[1], Aditya Rawal[2], Raymond J. Mooney[2], and Eduardo R. Hruschka[3]

[1] Department of ECE, University of Texas at Austin, USA
aacharya@utexas.edu
[2] Department of CS, University of Texas at Austin, USA
{aditya,mooney}@cs.utexas.edu
[3] Department of CS, University of São Paulo at São Carlos, Brazil
erh@icmc.usp.br

Abstract. This paper introduces two new frameworks, Doubly Supervised Latent Dirichlet Allocation (DSLDA) and its non-parametric variation (NP-DSLDA), that integrate two different types of supervision: topic labels and category labels. This approach is particularly useful for multitask learning, in which both latent and supervised topics are shared between multiple categories. Experimental results on both document and image classification show that both types of supervision improve the performance of both DSLDA and NP-DSLDA and that sharing both latent *and* supervised topics allows for better multitask learning.

1 Introduction

Humans can distinguish as many as 30,000 relevant object classes [7]. Training an isolated object detector for each of these different classes would require millions of training examples in aggregate. Computer vision researchers have proposed a more efficient learning mechanism in which object categories are learned via *shared* attributes, abstract descriptors of object properties such as "striped" or "has four legs" [17,25,24]. The attributes serve as an intermediate layer in a classifier cascade. The classifier in the first stage is trained to predict the attributes from the raw features and that in the second stage is trained to predict the categories from the attributes. During testing, only the raw features are observed and the attributes must be inferred. This approach is inspired by human perception and learning from high-level object descriptions. For example, from the phrase "eight-sided red traffic sign with white writing", humans can detect stop signs [25]. Similarly, from the description "large gray animals with long trunks", human can identify elephants. If the *shared* attributes transcend object class boundaries, such a classifier cascade is beneficial for *transfer learning* [28] where fewer labeled examples are available for some object categories compared to others [25].

Multitask learning (MTL) is a form of transfer learning in which simultaneously learning multiple related "tasks" allows each one to benefit from the learning of all of the others. If the tasks are related, training one task should provide helpful "inductive bias" for learning the other tasks. To enable the reuse of training information across multiple related tasks, all tasks might utilize the same latent shared intermediate representation – for example, a shared hidden layer in a multi-layer perceptron [11]. In this

H. Blockeel et al. (Eds.): ECML PKDD 2013, Part II, LNAI 8189, pp. 369–384, 2013.

case, the training examples for all tasks provide good estimates of the weights connecting the input layer to the hidden layer, and hence only a small number of examples per task is sufficient to achieve high accuracy. This approach is in contrast to "isolated" training of tasks where each task is learned independently using a separate classifier.

In this paper, our objective is to combine these two approaches to build an MTL framework that can use *both* attributes *and* class labels. The multiple tasks here correspond to different object categories (classes), and *both* observable attributes and latent properties are shared across the tasks. We want to emphasize that the proposed frameworks support general MTL; however, the datasets we use happen to be multiclass, where each class is treated as a separate "task" (as typical in multi-class learning based on binary classifiers). But, in no way are the frameworks restricted to multiclass MTL. Since attribute-based learning has been shown to support effective transfer learning in computer vision, the tasks here naturally correspond to object classes.

The basic building block of the frameworks presented in this paper is Latent Dirichlet Allocation (LDA) [9]. LDA focuses on unsupervised induction of multiple "topics" that help characterize a corpus of text documents. LDA has also been applied in computer vision where SIFT features are appropriately quantized to generate a *bag of visual words* for representing an image [35]. Since our experiments use both text and image data, we will overload the word "document" to denote either a text document or an image represented as a bag of visual words. The LDA approach has also been augmented to include two different types of supervision, document-level labels for either topics [31] or for an overall category inferred from the topics [43]. This paper introduces two new approaches, Doubly Supervised Latent Dirichlet Allocation (DSLDA) and its non-parametric variation (NP-DSLDA), that integrate both forms of supervision. At the topic level, the models assume that supervision is available for some topics during training (corresponding to the "attributes" used in computer vision), but that other topics remain latent (corresponding to the hidden layer in traditional MTL). The ability to provide supervision for *both* categories and a *subset* of topics improves the models' ability to perform accurate classification. In many applications, a variety of kinds of supervision may be naturally available from different sources at multiple levels of abstraction, such as keywords, topics, and categories for documents, or visual attribute, object, and scene labels for images. By effectively utilizing such multiple, interacting levels of supervision, DSLDA is able to learn more accurate predictors. In a supervised LDA [8,43] setting, forcing multiple tasks to share the same set of latent topics results in an LDA-based approach to MTL. By allowing supervision to also be provided for a subset of these shared topics, DSLDA and NP-DSLDA support a particularly effective form of MTL.

The rest of the paper is organized as follows. We present related literature in Section 2, followed by the descriptions of DSLDA and NP-DSLDA in Section 3 and Section 4 respectively. Experimental results on both multi-class image and document categorization are presented in Section 5, demonstrating the value of integrating both supervised and latent shared topics in diverse applications. Finally, future directions and conclusions are presented in Section 6.

Note on Notation: Vectors and matrices are denoted by bold-faced lowercase and capital letters, respectively. Scalar variables are written in italic font, and sets are denoted

by calligraphic uppercase letters. Dir(), Beta() and multinomial() stand for Dirichlet, Beta and multinomial distribution respectively.

2 Related Work

2.1 Statistical Topic Models

LDA [9] treats documents as a mixture of topics, which in turn are defined by a distribution over a set of words. The words in a document are assumed to be sampled from multiple topics. In its original formulation, LDA can be viewed as a purely-unsupervised form of dimensionality reduction and clustering of documents in the topic space, although several extensions of LDA have subsequently incorporated some sort of supervision. Some approaches provide supervision by labeling each document with its set of topics [31,32]. In particular, in *Labeled LDA* (LLDA [31]), the primary objective is to build a model of the words that indicate the presence of certain topic labels. For example, when a user explores a webpage based on certain tags, LLDA can be used to highlight interesting portions of the page or build a summary of the text from multiple webpages that share the same set of tags. The words in a given training document are assumed to be sampled *only* from the supervised topics, which the document has been labeled as covering.

Some other researchers [8,43,12] assume that supervision is provided for a single *response variable* to be predicted for a given document. The response variable might be real-valued or categorical, and modeled by a normal, Poisson, Bernoulli, multinomial or other distribution (see [12] for details). Some examples of documents with response variables are essays with their grades, movie reviews with their numerical ratings, web pages with their number of hits over a certain period of time, and documents with category labels. In *Maximum Entropy Discriminative LDA* (MedLDA) [43], the objective is to infer some low-dimensional (topic-based) representation of documents which is predictive of the response variable. Essentially, MedLDA solves two problems jointly – dimensionality reduction and max-margin classification using the features in the dimensionally-reduced space. Compared to earlier versions of supervised topic models [8,12], MedLDA has simpler update equations and produces superior experimental results. Therefore, in the frameworks presented in Sections 3.2 and 4, the max-margin principle adopted in MedLDA is preferred over other supervised topic models.

2.2 Transfer and Multitask Learning

Transfer learning allows the learning of some tasks to benefit the learning of others through either simultaneous [11] or sequential [10] training. In multitask learning (MTL [11]), a single model is simultaneously trained to perform multiple related tasks. MTL has emerged as a very promising research direction for various applications including biomedical informatics [6], marketing [15], natural language processing [2], and computer vision [34].

Many different MTL approaches have been proposed over the past 15 years (*e.g.*, see [38,28,29] and references therein). These include different learning methods, such

as empirical risk minimization using group-sparse regularizers [20,23,21], hierarchical Bayesian models [41,26] and hidden conditional random fields [30]. Evgeniou *et al.* [14] proposed the regularized MTL which constrained the models of all tasks to be close to each other. The task relatedness in MTL has also been modeled by constraining multiple tasks to share a common underlying structure [5,3,11]. Ando and Zhang [1] proposed a structural learning formulation, which assumed multiple predictors for different tasks shared a common structure on the underlying predictor space.

In all of the MTL formulations mentioned above, the basic assumption is that all tasks are related. In practical applications, these might not be the case and the tasks might exhibit a more sophisticated group structure. Such structure is handled using clustered multi-task learning (CMTL). In [4] CMTL is implemented by considering a mixture of Gaussians instead of single Gaussian priors. Xue *et al.* [39] introduced the Dirichlet process prior that automatically identifies subgroups of related tasks. In [19], a clustered MTL framework was proposed that simultaneously identified clusters and performed multi-task inference.

In the models presented in the next two sections, an LDA-based approach to MTL is easily obtained by maintaining a common set of topics to support the prediction of multiple response variables. This idea is analogous to implementing MTL using a common shared underlying structure [5,3,11]. We will also explain how NP-DSLDA is capable of performing CMTL.

3 Doubly Supervised LDA (DSLDA)

3.1 Task Definition

Assume we are given a training corpus consisting of N documents belonging to Y different classes (where each document belongs to exactly one class and each class corresponds to a different task). Further assume that each of these training documents is also annotated with a set of K_2 different topic "tags" (henceforth referred to as "supervised topics"). For computer vision data, the supervised topics correspond to the attributes provided by human experts. The objective is to train a model using the words in a data, as well as the associated supervised topic tags and class labels, and then use this model to classify completely unlabeled test data for which no topic tags nor class labels are provided. The human-provided supervised topics are presumed to provide abstract information that is helpful in predicting the class labels of test documents.

3.2 Generative Model

In order to include both types of supervision (class and topic labels), a combination of the approaches described in Section 2.1 is proposed. Note that LLDA uses *only* supervised topics and does not have any mechanism for generating class labels. On the other hand, MedLDA has only *latent* topics but learns a discriminative model for predicting classes from these topics. To the best of our knowledge, ours is the first LDA approach to integrate both types of supervision in a single framework. The generative process of DSLDA is described below.

For the n^{th} document, sample a topic selection probability vector $\boldsymbol{\theta}_n \sim \text{Dir}(\boldsymbol{\alpha}_n)$, where $\boldsymbol{\alpha}_n = \boldsymbol{\Lambda}_n \boldsymbol{\alpha}$ and $\boldsymbol{\alpha}$ is the parameter of a Dirichlet distribution of dimension K, which is the total number of topics. The topics are assumed to be of two types – latent and supervised, and there are K_1 latent topics and K_2 supervised topics. Therefore, $K = K_1 + K_2$. Latent topics are never observed, while supervised topics are observed in training but not in test data. Henceforth, in each vector or matrix with K components, it is assumed that the first K_1 components correspond to the latent topics and the next K_2 components to the supervised topics. $\boldsymbol{\Lambda}_n$ is a diagonal binary matrix of dimension $K \times K$. The k^{th} diagonal entry is unity if *either* $1 \leq k \leq K_1$ *or* $K_1 < k \leq K$ and the n^{th} document is tagged with the k^{th} topic. Also, $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2)$ where $\boldsymbol{\alpha}_1$ is a parameter of a Dirichlet distribution of dimension K_1 and $\boldsymbol{\alpha}_2$ is a parameter of a Dirichlet distribution of dimension K_2.

For the m^{th} word in the n^{th} document, sample a topic $z_{nm} \sim \text{multinomial}(\boldsymbol{\theta}'_n)$, where $\boldsymbol{\theta}'_n = (1-\epsilon)\{\boldsymbol{\theta}_{nk}\}_{k=1}^{k_1} \epsilon \{\Lambda_{n,kk}\boldsymbol{\theta}_{nk}\}_{k=1+k_1}^{K}$. This implies that the supervised topics are weighted by ϵ and the latent topics are weighted by $(1-\epsilon)$. Sample the word $w_{nm} \sim \text{multinomial}(\boldsymbol{\beta}_{z_{nm}})$, where $\boldsymbol{\beta}_k$ is a multinomial distribution over the vocabulary of words corresponding to the k^{th} topic.

For the n^{th} document, generate $Y_n = \arg\max_y \boldsymbol{r}_y^T \mathbb{E}(\bar{\mathbf{z}}_n)$ where Y_n is the class label associated with the n^{th} document, $\bar{\mathbf{z}}_n = \sum_{m=1}^{M_n} \mathbf{z}_{nm}/M_n$. Here, $\mathbf{z}_{nm}$ is an indicator vector of dimension K. $\boldsymbol{r}_y$ is a K-dimensional real vector corresponding to the y^{th} class, and it is assumed to have a prior distribution $\mathcal{N}(0, 1/C)$. M_n is the number of words in the n^{th} document. The maximization problem to generate Y_n (or the classification problem) is carried out using a max-margin principle.

Note that predicting each class is effectively treated as a separate task, and that the shared topics are useful for generalizing the performance of the model across classes. In particular, when all classes have few training examples, knowledge transfer between classes can occur through the shared topics. So, the mapping from the original feature space to the topic space is effectively learned using examples from all classes, and a few examples from each class are sufficient to learn the mapping from the reduced topic space to the class labels.

3.3 Inference and Learning

Let us denote the hidden variables by $\boldsymbol{Z} = \{\{z_{nm}\}, \{\boldsymbol{\theta}_n\}\}$, the observed variables by $\boldsymbol{X} = \{w_{nm}\}$ and the model parameters by $\boldsymbol{\kappa}_0$. The joint distribution of the hidden and observed variables is:

$$p(\mathbf{X}, \boldsymbol{Z}|\boldsymbol{\kappa}_0) = \prod_{n=1}^{N} p(\boldsymbol{\theta}_n|\boldsymbol{\alpha}_n) \prod_{m=1}^{M_n} p(z_{nm}|\boldsymbol{\theta}'_n) p(w_{nm}|\boldsymbol{\beta}_{\boldsymbol{z}_{nm}}) \tag{1}$$

To avoid computational intractability, inference and estimation are performed using Variational **EM**. The factorized approximation to the posterior distribution on hidden variables $\mathbf{Z}$ is given by:

$$q(\boldsymbol{Z}|\{\boldsymbol{\kappa}_n\}_{n=1}^{N}) = \prod_{n=1}^{N} q(\boldsymbol{\theta}_n|\boldsymbol{\gamma}_n) \prod_{m=1}^{M_n} q(z_{nm}|\boldsymbol{\phi}_{nm}), \tag{2}$$

where $\boldsymbol{\theta}_n \sim \text{Dir}(\boldsymbol{\gamma}_n)\ \forall n \in \{1, 2, \cdots, N\}$, $z_{nm} \sim \text{multinomial}(\boldsymbol{\phi}_{nm})\ \forall n \in \{1, 2, \cdots, N\}$ and $\forall m \in \{1, 2, \cdots, M_n\}$, and $\boldsymbol{\kappa}_n = \{\boldsymbol{\gamma}_n, \{\boldsymbol{\phi}_{nm}\}\}$, which is the set of variational parameters corresponding to the n^{th} instance. Further, $\boldsymbol{\gamma}_n = (\gamma_{nk})_{k=1}^K\ \forall n$, and $\boldsymbol{\phi}_{nm} = (\phi_{nmk})_{k=1}^K\ \forall n, m$. With the use of the lower bound obtained by the factorized approximation, followed by Jensen's inequality, DSLDA reduces to solving the following optimization problem[1]:

$$\begin{aligned} &\min_{q, \boldsymbol{\kappa}_0, \{\xi_n\}} \frac{1}{2}||\boldsymbol{r}||^2 - \mathcal{L}(q(\boldsymbol{Z})) + C\sum_{n=1}^{N}\xi_n, \\ &\text{s.t. } \forall n, y \neq Y_n : \mathbb{E}[\boldsymbol{r}^T \Delta f_n(y)] \geq 1 - \xi_n; \xi_n \geq 0. \end{aligned} \tag{3}$$

Here, $\Delta f_n(y) = f(Y_n, \bar{\boldsymbol{z}}_n) - f(y, \bar{\boldsymbol{z}}_n)$ and $\{\xi_n\}_{n=1}^N$ are the slack variables, and $f(y, \bar{\boldsymbol{z}}_n)$ is a feature vector whose components from $(y-1)K+1$ to yK are those of the vector $\bar{\boldsymbol{z}}_n$ and all the others are 0. $\mathbb{E}[\boldsymbol{r}^T \Delta f_n(y)]$ is the "expected margin" over which the true label Y_n is preferred over a prediction y. From this viewpoint, DSLDA projects the documents onto a combined topic space and then uses a max-margin approach to predict the class label. The parameter C penalizes the margin violation of the training data.

$$\begin{aligned} \phi^*_{nmk} \propto \Lambda_{n,kk} \exp\Big[&\psi(\gamma_{nk}) + \log(\beta_{kw_{nm}}) + \log(\epsilon') \\ &+ 1/M_n \sum_{y \neq Y_n} \mu_n(y)\mathbb{E}[r_{Y_n k} - r_{yk}]\Big]\ \forall n, m, k. \end{aligned} \tag{4}$$

$$\gamma^*_{nk} = \Lambda_{n,kk}\left[\alpha_k + \sum_{m=1}^{M_n}\phi_{nmk}\right]\ \forall n, vk. \tag{5}$$

$$\beta^*_{kv} \propto \sum_{n=1}^{N}\sum_{m=1}^{M_n}\phi_{nmk}\mathbb{I}_{\{w_{nm}=v\}}\ \forall k, v. \tag{6}$$

$$\begin{aligned} \mathcal{L}_{[\boldsymbol{\alpha}_1/\boldsymbol{\alpha}_2]} = &\left[\sum_{n=1}^{N}\log(\Gamma(\sum_{k=1}^{K}\alpha_{nk})) - \sum_{n=1}^{N}\sum_{k=1}^{K}\log(\Gamma(\alpha_{nk}))\right] \\ &+ \sum_{n=1}^{N}\sum_{k=1}^{K}\left[\psi(\gamma_{nk}) - \psi(\sum_{k=1}^{K}\gamma_{nk})\right](\alpha_{nk} - 1). \end{aligned} \tag{7}$$

Let $\mathcal{Q}$ be the set of all distributions having a fully factorized form as given in (2). Let the distribution q^* from the set $\mathcal{Q}$ optimize the objective in Eq. (3). The optimal values of corresponding variational parameters are given in Eqs. (4) and (5). In Eq. (4), $\epsilon' = (1-\epsilon)$ if $k \leq K_1$ and $\epsilon' = \epsilon$ otherwise. Since $\boldsymbol{\phi}_{nm}$ is a multinomial distribution, the updated values of the K components should be normalized to unity. The optimal values of $\boldsymbol{\phi}_{nm}$ depend on $\boldsymbol{\gamma_n}$ and vice-versa. Therefore, iterative optimization is adopted to maximize the lower bound until convergence is achieved.

[1] Please see [43] for further details.

During testing, one does not observe a document's supervised topics and, in principle, has to explore 2^{K_2} possible combinations of supervised tags – an expensive process. A simple approximate solution, as employed in LLDA [31], is to assume the absence of the variables $\{\boldsymbol{\Lambda}_n\}$ altogether in the test phase, and just treat the problem as inference in MedLDA with K latent topics. One can then threshold over the last K_2 topics if the tags of a test document need to be inferred. Equivalently, one can also assume $\boldsymbol{\Lambda}_n$ to be an identity matrix of dimension $K \times K \; \forall n$. This representation ensures that the expressions for update equations (4) and (5) do not change in the test phase.

In the M step, the objective in Eq. (3) is maximized w.r.t $\boldsymbol{\kappa}_0$. The optimal value of β_{kv} is given in Eq. (6). Since $\boldsymbol{\beta}_k$ is a multinomial distribution, the updated values of the V components should be normalized. However, numerical methods for optimization are required to update $\boldsymbol{\alpha}_1$ or $\boldsymbol{\alpha}_2$. The part of the objective function that depends on $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$ is given in Eq. (7). The update for the parameter $\boldsymbol{r}$ is carried out using a multi-class SVM solver [16]. With all other model and variational parameters held fixed (*i.e.* with $\mathcal{L}(q)$) held constant), the objective in Eq. (3) is optimized w.r.t. $\boldsymbol{r}$. A reader familiar with the updates in unsupervised LDA can see the subtle (but non-trivial) changes in the update equations for DSLDA.

4 Non-parametric DSLDA

We now propose a non-parametric extension of DSLDA (NP-DSLDA) that solves the model selection problem and automatically determines the best number of latent topics for modeling the given data. A modified stick breaking construction of Hierarchical Dirichlet Process (HDP) [33], recently introduced in [36] is used here which makes variational inference feasible. The idea in such representation is to share the corpus level atoms across documents by sampling atoms with replacement for each document and modifying the weights of these samples according to some other GEM distribution [33] whose parameter does not depend on the weights of the corpus-level atoms.

The combination of an infinite number of latent topics with a finite number of supervised topics in a single framework is not trivial and ours is the first model to accomplish this. One simpler solution is to introduce one extra binary hidden variable for each word in each document which could select either the set of latent topics or the set of supervised topics. Subsequently, a word in a document can be sampled from either the supervised or the latent topics based on the value sampled by the hidden "switching" variable. However, the introduction of such extra hidden variables adversely affects model performance as explained in [13]. In NP-DSLDA, we are able to avoid such extra hidden variables by careful modeling of the HDP. This will be evident in the generative process of NP-DSLDA presented below:

- Sample $\boldsymbol{\phi}_{k_1} \sim \text{Dir}(\boldsymbol{\eta}_1) \forall k_1 \in \{1, 2, \cdots, \infty\}$ and $\boldsymbol{\phi}_{k_2} \sim \text{Dir}(\boldsymbol{\eta}_2) \; \forall k_2 \in \{1, 2, \cdots, K_2\}$. $\boldsymbol{\eta}_1, \boldsymbol{\eta}_2$ are the parameters of Dirichlet distribution of dimension V.
- Sample $\beta'_{k_1} \sim \text{Beta}(1, \delta_0) \; \forall k_1 \in \{1, 2, \cdots, \infty\}$.
- For the n^{th} document, sample $\boldsymbol{\pi}_n^{(2)} \sim \text{Dir}(\boldsymbol{\Lambda}_n \boldsymbol{\alpha}_2)$. $\boldsymbol{\alpha}_2$ is the parameter of Dirichlet of dimension K_2. $\boldsymbol{\Lambda}_n$ is a diagonal binary matrix of dimension $K_2 \times K_2$. The k^{th} diagonal entry is unity if the n^{th} word is tagged with the k^{th} supervised topic.

- $\forall n, \forall t \in \{1, 2, \cdots, \infty\}$, sample $\pi'_{nt} \sim \text{Beta}(1, \alpha_0)$. Assume $\boldsymbol{\pi}_n^{(1)} = (\pi_{nt})_t$ where $\pi_{nt} = \pi'_{nt} \prod_{l<t}(1 - \pi'_{nl})$.
- $\forall n, \forall t$, sample $c_{nt} \sim \text{multinomial}(\boldsymbol{\beta})$ where $\beta_{k_1} = \beta'_{k_1} \prod_{l<k_1}(1 - \beta'_l)$. $\boldsymbol{\pi}_n^{(1)}$ represents the probability of selecting the sampled atoms in $\boldsymbol{c}_n$. Due to sampling with replacement, $\boldsymbol{c}_n$ can contain multiple atoms of the same index from the corpus level DP.
- For the m^{th} word in the n^{th} document, sample $z_{nm} \sim \text{multinomial}((1-\epsilon)\boldsymbol{\pi}_n^{(1)}, \epsilon\boldsymbol{\pi}_n^{(2)})$. This implies that w.p. ϵ, a topic is selected from the set of supervised topics and w.p. $(1-\epsilon)$, a topic is chosen from the set of (infinite number of) unsupervised topics. Note that by weighting the $\boldsymbol{\pi}$'s appropriately, the need for additional hidden "switching" variable is avoided.
- Sample w_{nm} from a multinomial given by the following equation:

$$\prod_{k_1=1}^{\infty} \prod_{v=1}^{V} \phi_{k_1 v}^{\mathbb{I}\{w_{nm}=v\}\mathbb{I}\{c_{nz_{nm}}=k_1 \in \{1,\cdots,\infty\}\}} \prod_{k_2=1}^{K_2} \prod_{v=1}^{V} \phi_{k_2 v}^{\mathbb{I}\{w_{nm}=v\}\mathbb{I}\{z_{nm}=k_2 \in \{1,\cdots,K_2\}\}} \quad (8)$$

The joint distribution of NP-DSLDA is given as follows:

$$p(\mathbf{X}, \boldsymbol{Z}|\boldsymbol{\kappa}_0) = \prod_{k_1=1}^{\infty} p(\phi_{k_1}|\boldsymbol{\eta}_1) p(\beta'_{k_1}|\boldsymbol{\delta}_0) \prod_{k_2=1}^{K_2} p(\phi_{k_2}|\boldsymbol{\eta}_2) \prod_{n=1}^{N} p(\boldsymbol{\pi}_n^{(2)}|\boldsymbol{\alpha}_2) \quad (9)$$

$$\prod_{t=1}^{\infty} p(\boldsymbol{\pi}_{nt}^{\prime(1)}|\boldsymbol{\alpha}_0) p(c_{nt}|\boldsymbol{\beta}') \prod_{m=1}^{M_n} p(z_{nm}|\boldsymbol{\pi}_n^{(1)}, \boldsymbol{\pi}_n^{(2)}, \epsilon) p(w_{nm}|\boldsymbol{\phi}, c_{nz_{nm}}, z_{nm}).$$

As an approximation to the posterior distribution over the hidden variables, we use the following factorized distribution:

$$q(\boldsymbol{Z}|\boldsymbol{\kappa}) = \prod_{k_1=1}^{\overline{K_1}} q(\phi_{k_1}|\boldsymbol{\lambda}_{k_1}) \prod_{k_2=1}^{K_2} q(\phi_{k_2}|\boldsymbol{\lambda}_{k_2}) \prod_{k_1=1}^{\overline{K_1}-1} q(\beta'_{k_1}|u_{k_1}, v_{k_1}) \quad (10)$$

$$\prod_{n=1}^{N} q(\boldsymbol{\pi}_n^{(2)}|\boldsymbol{\gamma}_n) \prod_{t=1}^{T-1} q(\boldsymbol{\pi}_{nt}^{\prime(1)}|a_{nt}, b_{nt}) \prod_{t=1}^{T} q(c_{nt}|\boldsymbol{\varphi}_{nt}) \prod_{m=1}^{M_n} q(z_{nm}|\boldsymbol{\zeta}_{nm}).$$

Here, $\boldsymbol{\kappa}_0$ and $\boldsymbol{\kappa}$ denote the sets of model and variational parameters, respectively. $\overline{K_1}$ is the truncation limit of the corpus-level Dirichlet Process and T is the truncation limit of the document-level Dirichlet Process. $\{\boldsymbol{\lambda}_k\}$ are the parameters of Dirichlet each of dimension V. $\{u_{k_1}, v_{k_1}\}$ and $\{a_{nt}, b_{nt}\}$ are the parameters of variational Beta distribution corresponding to corpus level and document level sticks respectively. $\{\boldsymbol{\varphi}_{nt}\}$ are multinomial parameters of dimension $\overline{K_1}$ and $\{\boldsymbol{\zeta}_{nm}\}$ are multinomials of dimension $(T + K_2)$. $\{\boldsymbol{\gamma}_n\}_n$ are parameters of Dirichlet distribution of dimension K_2.

The underlying optimization problem takes the same form as in Eq. (3). The only difference lies in the calculation of $\Delta f_n(y) = f(Y_n, \bar{\boldsymbol{s}}_n) - f(y, \bar{\boldsymbol{s}}_n)$. The first set of dimensions of $\bar{\boldsymbol{s}}_n$ (corresponding to the unsupervised topics) is given by $1/M_n \sum_{m=1}^{M_n} \boldsymbol{c}_{nz_{nm}}$, where $\boldsymbol{c}_{nt}$ is an indicator vector over the set of unsupervised topics. The following K_2 dimensions (corresponding to the supervised topics) are given by $1/M_n \sum_{m=1}^{M_n} \boldsymbol{z}_{nm}$. After the variational approximation with $\overline{K_1}$ number of corpus level sticks, $\bar{\boldsymbol{s}}_n$ turns out

to be of dimension $(\overline{K_1}+K_2)$ and the feature vector $f(y,\bar{\boldsymbol{s}}_n)$ constitutes $Y(\overline{K_1}+K_2)$ elements. The components of $f(y,\bar{\boldsymbol{s}}_n)$ from $(y-1)(\overline{K_1}+K_2)+1$ to $y(\overline{K_1}+K_2)$ are those of the vector $\bar{\boldsymbol{s}}_n$ and all the others are 0. Essentially, due to the variational approximation, NP-DSLDA projects each document on to a combined topic space of dimension $(\overline{K_1}+K_2)$ and learns the mapping from this space to the classes.

$$\zeta^*_{nmt} \propto \exp\Bigg[[\psi(a_{nt})-\psi(a_{nt}+b_{nt})]\mathbb{I}_{\{t<T\}}+\sum_{t'=1}^{t-1}[\psi(b_{nt'})-\psi(a_{nt'}+b_{nt'})] \tag{11}$$
$$+\sum_{k_1=1}^{\overline{K_1}}\varphi_{ntk_1}\left[\psi(\lambda_{k_1 w_{nm}})-\psi(\sum_{v=1}^{V}\lambda_{k_1 v})\right]$$
$$+\sum_{y\neq Y_n}\mu_n(y)\sum_{k_1=1}^{\overline{K_1}}\mathbb{E}[r_{Y_n k_1}-r_{yk_1}]\varphi_{ntk_1}\Bigg]\ \forall n,m,t.$$

$$\zeta^*_{nm(T+k_2)} \propto \Lambda_{nk_2k_2}\exp\Bigg[\psi(\gamma_{nk_2})-\psi(\sum_{k_2=1}^{K_2}\gamma_{nk_2})+\psi(\lambda_{(\overline{K_1}+k_2)w_{nm}}) \tag{12}$$
$$-\psi(\sum_{v=1}^{V}\lambda_{(\overline{K_1}+k_2)v})+1/M_n\sum_{y\neq Y_n}\mu_n(y)\mathbb{E}[r_{Y_n(\overline{K_1}+k_2)}-r_{y(\overline{K_1}+k_2)}]\Bigg]\ \forall n,m,k_2.$$

$$\varphi^*_{ntk_1} \propto \exp\Big[[\psi(u_{k_1})-\psi(u_{k_1}+v_{k_1})]\,\mathbb{I}_{\{k_1<K_1\}} \tag{13}$$
$$+\sum_{k'=1}^{k_1-1}[\psi(v_{k'})-\psi(u_{k'}+v_{k'})]+\sum_{m=1}^{M_n}\zeta_{nmt}\left[\psi(\lambda_{k_1 w_{nm}})-\psi(\sum_{v=1}^{V}\lambda_{k_1 v})\right]$$
$$+1/M_n\sum_{y\neq Y_n}\mu_n(y)\mathbb{E}[r_{Y_n k_1}-r_{yk_1}]\left(\sum_{m=1}^{M_n}\zeta_{nmt}\right)\Bigg]\ \forall n,t,k_1.$$

Some of the update equations of NP-DSLDA are given in the above equations, where $\{\varphi_{ntk_1}\}$ are the set of variational parameters that characterize the assignment of the documents to the global set of $(\overline{K_1}+K_2)$ topics. One can see how the effect of the class labels is included in the update equation of $\{\varphi_{ntk_1}\}$ via the average value of the parameters $\{\zeta_{nmt}\}$. This follows intuitively from the generative assumption. update exists for the model parameters and hence numerical optimization has to be used. Other updates are either similar to DSLDA or the model in [36] and are omitted due to space constraints. $\{\boldsymbol{\zeta}_{nm}\}$, corresponding to supervised and unsupervised topics, should be individually normalized and then scaled by ϵ and $(1-\epsilon)$ respectively. Otherwise, the effect of the Dirichlet prior on supervised topics will get compared to that of the GEM prior on the unsupervised topics which does not follow the generative assumptions. The variational parameters $\{\boldsymbol{\lambda}_k\}$ and $\{\boldsymbol{\varphi}_{nt}\}$ are also normalized.

Note that NP-DSLDA offers some flexibility with respect to the latent topics that could be dominant for a specific task. One could therefore postulate that NP-DSLDA can learn the clustering of tasks from the data itself by making a subset of latent topics to be dominant for a set of tasks. Although do not have supporting experiments, NP-DSLDA is capable in principle of performing clustered multi-task learning without any prior assumption on the relatedness of the tasks.

5 Experimental Evaluation

5.1 Data Description

Our evaluation used two datasets, a text corpus and a multi-class image database, as described below.

aYahoo Data. The first set of experiments was conducted with the aYahoo image dataset from [17] which has 12 classes – carriage, centaur, bag, building, donkey, goat, jetski, monkey, mug, statue, wolf, and zebra.[2] Each image is annotated with relevant visual attributes such as "has head", "has wheel", "has torso" and 61 others, which we use as the supervised topics. Using such intermediate "attributes" to aid visual classification has become a popular approach in computer vision [25,24]. After extracting SIFT features [27] from the raw images, quantization into 250 clusters is performed, defining the vocabulary for the bag of visual words. Images with less than two attributes were discarded. The resulting dataset of size 2,275 was equally split into training and test data.

ACM Conference Data. The text corpus consists of conference paper abstracts from two groups of conferences. The first group has four conferences related to data mining – WWW, SIGIR, KDD, and ICML, and the second group consists of two VLSI conferences – ISPD and DAC. The classification task is to determine the conference at which the abstract was published. As supervised topics, we use keywords provided by the authors, which are presumably useful in determining the conference venue. Since authors usually take great care in choosing keywords so that their paper is retrieved by relevant searches, we believed that such keywords made a good choice of supervised topics. Part of the data, crawled from ACM's website, was used in [37]. A total of 2,300 abstracts were collected each of which had at least three keywords and an average of 78 (± 33.5) words. After stop-word removal, the vocabulary size for the assembled data is 13,412 words. The final number of supervised topics, after some standard pre-processing of keywords, is 55. The resulting dataset was equally split into training and test data.

5.2 Methodology

In order to demonstrate the contribution of each aspect of the overall model, DSLDA and NP-DSLDA are compared against the following simplified models:

[2] `http://vision.cs.uiuc.edu/attributes/`

- MedLDA with **o**ne-**v**s-**a**ll classification (MedLDA-OVA): A separate model is trained for each class using a one-vs-all approach leaving no possibility of transfer across classes.
- MedLDA with **m**ulti**t**ask **l**earning (MedLDA-MTL): A single model is learned for all classes where the latent topics are shared across classes.
- DSLDA with **o**nly **s**hared **s**upervised **t**opics (DSLDA-OSST): A model in which supervised topics are used and shared across classes but there are no latent topics.
- DSLDA with **n**o **s**hared **l**atent **t**opics (DSLDA-NSLT): A model in which only supervised topics are shared across classes and a separate set of latent topics is maintained for each class.
- **M**ajority **c**lass **m**ethod (MCM): A simple baseline which always picks the most common class in the training data.

These baselines are useful for demonstrating the utility of *both* supervised and latent shared topics for multitask learning in DSLDA. MedLDA-OVA is a non-transfer method, where a separate model is learned for each of the classes, *i.e.* one of the many classes is considered as the positive class and the union of the remaining ones is treated as the negative class. Since the models for each class are trained separately, there is no possibility of sharing inductive information across classes. MedLDA-MTL trains on examples from all classes simultaneously, and thus allows for sharing of inductive information *only* through a common set of latent topics. In DSLDA-OSST, only supervised topics are maintained and knowledge transfer can *only* take place via these supervised topics. DSLDA-NSLT uses shared supervised topics but also includes latent topics which are *not* shared across classes. This model provides for transfer *only* through shared supervised topics but provides extra modeling capacity compared to DSLDA-OSST through the use of latent topics that are not shared. DSLDA and NP-DSLDA are MTL frameworks where both supervised *and* latent topics are shared across all classes. Note that, all of the baselines can be implemented using DSLDA with a proper choice of $\boldsymbol{\Lambda}$ and ϵ. For example, DSLDA-OSST is just a special case of DSLDA with ϵ fixed at 1.

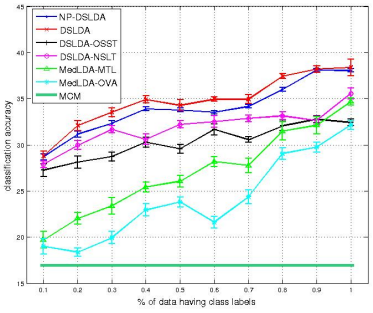

Fig. 1. $p_1 = 0.5$ (aYahoo)

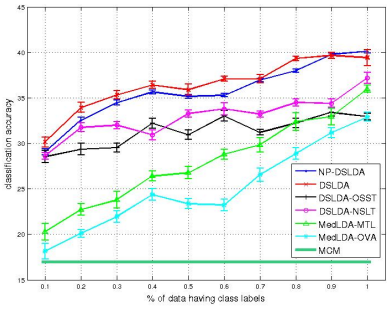

Fig. 2. $p_1 = 0.7$ (aYahoo)

Table 1. Illustration of Latent and Supervised Topics

LT1	function, label, graph, classification, database, propagation, algorithm, accuracy, minimization, transduction
LT2	performance, design, processor, layer, technology, device, bandwidth, architecture, stack, system
CAD	design, optimization, mapping, pin, simulation, cache, programming, routing, biochip, electrode
VLSI	design, physical, lithography, optimization, interdependence, global, robust, cells, layout, growth
IR	algorithm, web, linear, query, precision, document, repair, site, search, semantics
Ranking	integration, catalog, hierarchical, dragpushing, structure, source, sequence, alignment, transfer, flattened, speedup
Learning	model, information, trajectory, bandit, mixture, autonomous, hierarchical, feedback, supervised, task

In order to explore the effect of different amounts of both types of supervision, we varied the amount of both topic-level and class-level supervision. Specifically, we provided topic supervision for a fraction, p_1, of the overall training set, and then provided class supervision for only a further fraction p_2 of this data. Therefore, only $p_1 * p_2$ of the overall training data has class supervision. By varying the number of latent topics from 20 to 200 in steps of 10, we found that $K_1 = 100$ generally worked the best for all the parametric models. Therefore, we show parametric results for 100 latent topics. For each combination of (p_1, p_2), 50 random trials were performed with $C = 10$. To maintain equal representational capacity, the total number of topics K is held the same across all parametric models (except for DSLDA-OSST where the total number of topics is K_2). For NP-DSLDA, following the suggestion of [36], we set $\overline{K_1} = 150$ and $T = 40$, which produced uniformly good results. When required, ϵ was chosen using 5-fold internal cross-validation using the training data.

5.3 Results

Figs. 1 and 2 present representative learning curves for the image data, showing how classification accuracy improves as the amount of class supervision (p_2) is increased. Results are shown for two different amounts of topic supervision ($p_1 = 0.5$ and $p_1 = 0.7$). Figs. 3 and 4 present similar learning curves for the text data. The error bars in the curves show standard deviations across the 50 trials.

The results demonstrate that DSLDA and NP-DSLDA quite consistently outperform all of the baselines, clearly demonstrating the advantage of combining both types of

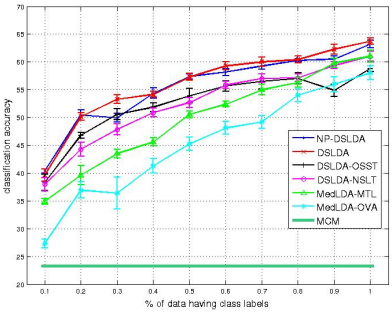

Fig. 3. $p_1 = 0.5$ (Conference)

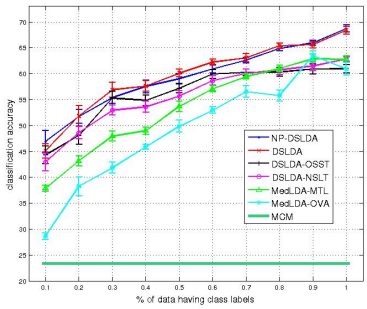

Fig. 4. $p_1 = 0.7$ (Conference)

topics. NP-DSLDA performs about as well as DSLDA, for which the optimal number of latent topics has been chosen using an expensive model-selection search. This demonstrates that NP-DSLDA is doing a good job of automatically selecting an appropriate number of latent topics.

Overall, DSLDA-OSST and MedLDA-MTL perform about the same, showing that, individually, both latent and supervised shared topics each support multitask learning about equally well when used alone. However, combining both types of topics provides a clear improvement.

MedLDA-OVA performs quite poorly when there is only a small amount of class supervision (note that this baseline uses *only* class labels). However, the performance approaches the others as the amount of class supervision increases. This is consistent with the intuition that multitask learning is most beneficial when each task has limited supervision and therefore has more to gain by sharing information with other tasks.

Shared supervised topics clearly increase classification accuracy when class supervision is limited (i.e. small values of p_2), as shown by the performance of both DSLDA-NSLT and DSLDA-OSST. When $p_2 = 1$ (equal amounts of topic and class supervision), DSLDA-OSST, MedLDA-MTL and MedLDA-OVA all perform similarly; however, by exploiting *both* types of supervision, DSLDA and NP-DSLDA still maintain a performance advantage.

5.4 Topic Illustration

In Table 1, we show the most indicative words for several topics discovered by DSLDA from the text data (with $p_1 = 0.8$ and $p_2 = 1$). LT1 and LT2 correspond to the most frequent latent topics assigned to documents in the two broad categories of conferences (data mining and VLSI, respectively). The other five topics are supervised ones. CAD and IR stand for Computer Aided Design and Information Retrieval respectively. The illustrated topics are particularly discriminative when classifying documents.

5.5 Discussion

DSLDA-NSLT only allows sharing of supervised topics and its implementation is not straightforward. Since MedLDA-OVA, MedLDA-MTL and DSLDA use K topics (latent or a combination of supervised and latent), to make the comparison fair, it is necessary to maintain the same number of topics for DSLDA-NSLT. This ensures that the models compared have the same representational capacity. Therefore, for each class in DSLDA-NSLT, k_2/Y latent topics are maintained. While training DSLDA-NSLT with examples from the y^{th} class, only a subset of the first k_1 topics (or a subset of the supervised ones based on which of them are present in the training documents) and the next $\left(\frac{(y-1)k_2}{Y} + 1\right)^{\text{th}}$ to $\left(\frac{yk_2}{Y}\right)^{\text{th}}$ topics are considered to be "active" among the latent topics. The other latent topics are assumed to have zero contribution, implying that the parameters associated with these topics are not updated based on observations of documents belonging to class y. During testing, however, one needs to project a document onto the entire K-dimensional space, and the class label is predicted based on this representation and the parameters $\boldsymbol{r}$.

Overall, the results support the hypothesis that DSLDA's ability to incorporate both supervised and latent topics allow it to achieve better predictive performance compared to baselines that exploit only one, the other, or neither. Furthermore, NP-DSLDA is able to automate model-selection, performing nearly as well as DSLDA with optimally chosen parameters.

6 Future Work and Conclusion

This paper has introduced Doubly Supervised LDA (DSLDA) and non-parametric DSLDA (NP-DSLDA), novel approaches that combine the following – generative and discriminative models, latent and supervised topics, and class and topic level supervision, in a principled probabilistic manner. Four ablations of this model are also evaluated in order to understand the individual effects of latent/supervised topics and multitask learning on the overall model performance. The general idea of "double supervision" could be applied to many other models, for example, in multi-layer perceptrons, latent SVMs [40] or in deep belief networks [18]. In MTL, sharing tasks blindly is not always a good approach and further extension with clustered MTL [42] is possible. Based on a very recent study [22], a sampling based algorithm could also be developed for NP-DSLDA, possibly leading to even better performance.

Acknowledgments. This research was partially supported by ONR ATL Grant N00014-11-1-0105, NSF Grants (IIS-0713142 and IIS-1016614) and by the Brazilian Research Agencies FAPESP and CNPq.

References

1. Ando, R., Zhang, T.: A framework for learning predictive structures from multiple tasks and unlabeled data. Journal of Machine Learning Research 6, 1817–1853 (2005)
2. Ando, R.K.: Applying alternating structure optimization to word sense disambiguation. In: Proceedings of Computational Natural Language Learning (2006)
3. Argyriou, A., Micchelli, C.A., Pontil, M., Ying, Y.: A spectral regularization framework for multi-task structure learning. In: Proceedings of Neural Information Processing Systsems (2007)
4. Bakker, B., Heskes, T.: Task clustering and gating for Bayesian multitask learning. Journal of Machine Learning Research 4 (2003)
5. Ben-David, S., Schuller, R.: Exploiting task relatedness for multiple task learning. In: Schölkopf, B., Warmuth, M.K. (eds.) COLT/Kernel 2003. LNCS (LNAI), vol. 2777, pp. 567–580. Springer, Heidelberg (2003)
6. Bickel, S., Bogojeska, J., Lengauer, T., Scheffer, T.: Multi-task learning for HIV therapy screening. In: Proceedings of International Conference on Machine Learning, pp. 56–63. ACM, New York (2008)
7. Biederman, I.: Recognition-by-components: A theory of human image understanding. Psychological Review 94, 115–147 (1987)
8. Blei, D.M., Mcauliffe, J.D.: Supervised topic models. In: Proceedings of Neural Information Processing Systems (2007)

9. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. Journal of Machine Learning Research 3, 993–1022 (2003)
10. Bollacker, K.D., Ghosh, J.: Knowledge transfer mechanisms for characterizing image datasets. In: Soft Computing and Image Processing. Physica-Verlag, Heidelberg (2000)
11. Caruana, R.: Multitask learning. Machine Learning 28, 41–75 (1997)
12. Chang, J., Blei, D.: Relational topic models for document networks. In: Proceedings of Artificial Intelligence and Statistics (2009)
13. Eisenstein, J., Ahmed, A., Xing, E.P.: Sparse additive generative models of text. In: Proceedings of International Conference on Machine Learning, pp. 1041–1048 (2011)
14. Evgeniou, T., Micchelli, C.A., Pontil, M.: Learning multiple tasks with kernel methods. Journal of Machine Learning Research 6, 615–637 (2005)
15. Evgeniou, T., Pontil, M., Toubia, O.: A convex optimization approach to modeling consumer heterogeneity in conjoint estimation. Marketing Science 26(6), 805–818 (2007)
16. Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., Lin, C.-J.: LIBLINEAR: A library for large linear classification. Journal of Machine Learning Research 9, 1871–1874 (2008)
17. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. In: Proceedings of Computer Vision and Pattern Recognition (2009)
18. Hinton, G.E., Osindero, S.: A fast learning algorithm for deep belief nets. Neural Computation 18, 2006 (2006)
19. Jacob, L., Bach, F., Vert, J.-P.: Clustered multi-task learning: A convex formulation. CoRR, abs/0809.2085 (2008)
20. Jalali, A., Ravikumar, P., Sanghavi, S., Ruan, C.: A Dirty Model for Multi-task Learning. In: Proceedings of Neural Information Processing Systems (December 2010)
21. Jenatton, R., Audibert, J., Bach, F.: Structured variable selection with sparsity-inducing norms. Journal of Machine Learning Research 12, 2777–2824 (2011)
22. Jiang, Q., Zhu, J., Sun, M., Xing, E.: Monte carlo methods for maximum margin supervised topic models. In: Proceedings of Neural Information Processing Systems, pp. 1601–1609 (2012)
23. Kim, S., Xing, E.P.: Tree-guided group lasso for multi-task regression with structured sparsity. In: Proceedings of International Conference on Machine Learning, pp. 543–550 (2010)
24. Kovashka, A., Vijayanarasimhan, S., Grauman, K.: Actively selecting annotations among objects and attributes. In: International Conference on Computer Vision, pp. 1403–1410. IEEE (2011)
25. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by betweenclass attribute transfer. In: Proceedings of Computer Vision and Pattern Recognition (2009)
26. Low, Y., Agarwal, D., Smola, A.J.: Multiple domain user personalization. In: Proceedings of Knowledge Discovery and Data Mining, pp. 123–131 (2011)
27. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
28. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering 22, 1345–1359 (2010)
29. Passos, A., Rai, P., Wainer, J., Daumé III, H.: Flexible modeling of latent task structures in multitask learning. In: Proceedings of International Conference on Machine Learning (2012)
30. Quattoni, A., Wang, S., Morency, L.P., Collins, M., Darrell, T., Csail, M.: Hidden-state conditional random fields. In: IEEE Transactions on Pattern Analysis and Machine Intelligence (2007)
31. Ramage, D., Hall, D., Nallapati, R., Manning, C.D.: Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora. In: Proceedings of Empirical Methods in Natural Language Processing, pp. 248–256 (2009)

32. Rubin, T.N., Chambers, A., Smyth, P., Steyvers, M.: Statistical topic models for multi-label document classification. CoRR, abs/1107.2462 (2011)
33. Teh, Y.W., Jordan, M.I., Beal, M.J., Blei, D.M.: Hierarchical Dirichlet Processes. Journal of the American Statistical Association 101, 1566–1581 (2006)
34. Torralba, A., Murphy, K.P., Freeman, W.T.: Sharing visual features for multiclass and multiview object detection. IEEE Trans. Pattern Anal. Mach. Intell. 29(5), 854–869 (2007)
35. Wang, C., Blei, D.M., Li, F.F.: Simultaneous image classification and annotation. In: Proceedings of Computer Vision and Pattern Recognition, pp. 1903–1910 (2009)
36. Wang, C., Paisley, J.W., Blei, D.M.: Online variational inference for the hierarchical Dirichlet process. Journal of Machine Learning Research - Proceedings Track 15, 752–760 (2011)
37. Wang, C., Thiesson, B., Meek, C., Blei, D.: Markov topic models. In: Proceedings of Artificial Intelligence and Statistics (2009)
38. Weinberger, K., Dasgupta, A., Langford, J., Smola, A., Attenberg, J.: Feature hashing for large scale multitask learning. In: Proceedings of International Conference on Machine Learning, pp. 1113–1120 (2009)
39. Xue, Y., Liao, X., Carin, L., Krishnapuram, B.: Multi-task learning for classification with Dirichlet process priors. Journal of Machine Learning Research 8, 35–63 (2007)
40. Yu, C.J., Joachims, T.: Learning structural SVMs with latent variables. In: Proceedings of International Conference on Machine Learning, pp. 1169–1176 (2009)
41. Zhang, J., Ghahramani, Z., Yang, Y.: Flexible latent variable models for multi-task learning. Machine Learning 73(3), 221–242 (2008)
42. Zhou, J., Chen, J., Ye, J.: Clustered Multi-Task Learning Via Alternating Structure Optimization. In: Proceedings of Neural Information Processing Systems (2011)
43. Zhu, J., Ahmed, A., Xing, E.P.: MedLDA: maximum margin supervised topic models for regression and classification. In: Proceedings of International Conference on Machine Learning, pp. 1257–1264 (2009)

Active Multitask Learning with Doubly Supervised Latent Dirichlet Allocation

Anonymous Author(s)
Affiliation
Address
email

Abstract

This paper introduces two models – Doubly Supervised Latent Dirichlet Allocation (DSLDA) that makes use of both shared latent and *supervised* topics to accomplish multitask learning (MTL) and Active Doubly Supervised Latent Dirichlet Allocation (Act-DSLDA) that integrates MTL and active learning in the same framework. Experimental results on both document and image classification show that integrating MTL and active learning along with shared latent and supervised topics is superior to other methods which do not use all of these components.

1 Introduction

Research in computer vision for designing an automated object detector has primarily been focused on *either* gathering large datasets of web images [1, 2] *or* by formulating new algorithms that can reduce the degree of human intervention in the learning process. In one of the learning methodologies, *shared* attributes, abstract descriptors of object properties [3, 4, 5] are used to serve as an intermediate layer in a classifier cascade. If the *shared* attributes transcend object class boundaries, such a classifier cascade is beneficial for *transfer learning* [6]. Another group of researchers have formulated methods based on active learning for reducing the expense of human annotations where the system can request labels for the most informative examples [7, 4]. In this paper, our objective is to combine these two orthogonal approaches in order to leverage the benefits of both – learning from a shared feature space and making active queries.

2 Related Work

Unsupervised LDA has been extended to account for supervision. In *Labeled LDA* (LLDA [8]), the primary objective is to build a model of the words that indicate the presence of certain topic labels. Some other researchers [9, 10, 11] assume that supervision is provided for a single *response variable* to be predicted for a given document. In *Maximum Entropy Discriminative LDA* (MedLDA) [10], the objective is to infer some low-dimensional (topic-based) representation of documents which is predictive of the response variable. Transfer learning allows the learning of some tasks to benefit the learning of others through either simultaneous [12] or sequential [13] training. In multitask learning (MTL [12]), a single model is simultaneously trained to perform multiple related tasks (*e.g.*, see [6, 14]). Finally, there has been some effort to integrate active and transfer learning in the same framework [15, 16, 17] . However, none of these approaches deal with topic models or query over both supervised topic labels and class labels

3 Doubly Supervised Latent Dirichlet Allocation (DSLDA)

Assume we are given a training corpus consisting of N documents belonging to Y different classes (where each document belongs to exactly one class and each class corresponds to a different task). Further assume that each of these training documents is also annotated with a set of K_2 different topic "tags" (henceforth referred to as "supervised topics"). The objective is to train a model using the words in a data, as well as the associated supervised topic tags and class labels, and then use this model to classify completely unlabeled test data for which no topic tags nor class labels are provided. The DSLDA model is now described below.

• For the n^{th} document, sample a topic selection probability vector $\boldsymbol{\theta}_n \sim \text{Dir}(\boldsymbol{\alpha}_n)$, where $\boldsymbol{\alpha}_n = \boldsymbol{\Lambda}_n \boldsymbol{\alpha}$ and $\boldsymbol{\alpha}$ is the parameter of a Dirichlet distribution of dimension K, which is the total number of topics. The topics are assumed to be of two types – latent and supervised, and there are K_1 latent topics and K_2 supervised topics. Therefore, $K = K_1 + K_2$. Latent topics are never observed, while supervised topics are observed in training but not in test data. Henceforth, in each vector or matrix with K components, it is assumed that the first K_1 components correspond to the latent topics and the next K_2 components to the supervised topics. $\boldsymbol{\Lambda}_n$ is a diagonal binary matrix of dimension $K \times K$. The k^{th} diagonal entry is unity if *either* $1 \leq k \leq K_1$ *or* $K_1 < k \leq K$ and the n^{th} document is tagged with the k^{th} topic. Also, $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2)$ where $\boldsymbol{\alpha}_1$ is a parameter of a Dirichlet distribution of dimension K_1 and $\boldsymbol{\alpha}_2$ is a parameter of a Dirichlet distribution of dimension K_2.

• For the m^{th} word in the n^{th} document, sample a topic $z_{nm} \sim \text{multinomial}(\boldsymbol{\theta}'_n)$, where $\boldsymbol{\theta}'_n = (1-\epsilon)\{\boldsymbol{\theta}_{nk}\}_{k=1}^{k_1} \epsilon\{\Lambda_{n,kk}\boldsymbol{\theta}_{nk}\}_{k=1+k_1}^{K}$. This implies that the supervised topics are weighted by ϵ and the latent topics are weighted by $(1-\epsilon)$. Sample the word $w_{nm} \sim \text{multinomial}(\boldsymbol{\beta}_{z_{nm}})$, where $\boldsymbol{\beta}_k$ is a multinomial distribution over the vocabulary of words corresponding to the k^{th} topic.

• For the n^{th} document, generate $Y_n = \arg\max_y \boldsymbol{r}_y^T \mathbb{E}(\bar{\mathbf{z}}_n)$ where Y_n is the class label associated with the n^{th} document, $\bar{\mathbf{z}}_n = \sum_{m=1}^{M_n} \mathbf{z}_{nm}/M_n$. Here, $\mathbf{z}_{nm}$ is an indicator vector of dimension K. $\boldsymbol{r}_y$ is a K-dimensional real vector corresponding to the y^{th} class, and it is assumed to have a prior distribution $\mathcal{N}(0, 1/C)$. M_n is the number of words in the n^{th} document. The maximization problem to generate Y_n (or the classification problem) is carried out using a max-margin principle.

Note that predicting each class is effectively treated as a separate task, and that the shared topics are useful for generalizing the performance of the model across classes. In particular, when all classes have few training examples, knowledge transfer between classes can occur through the shared topics.

Let us denote the hidden variables by $\boldsymbol{Z} = \{\{z_{nm}\}, \{\boldsymbol{\theta}_n\}\}$, the observed variables by $\boldsymbol{X} = \{w_{nm}\}$ and the model parameters by $\boldsymbol{\kappa}_0$. To avoid computational intractability, inference and estimation are performed using Variational **EM** using a completely factorzied approximation $q(\mathbf{Z})$. With the use of the lower bound obtained by the factorized approximation, followed by Jensen's inequality, DSLDA reduces to solving the following optimization problem[1]:

$$\min_{q,\boldsymbol{\kappa}_0,\{\xi_n\}} \frac{1}{2}||\boldsymbol{r}||^2 - \mathcal{L}(q(\boldsymbol{Z})) + C\sum_{n=1}^{N}\xi_n, \text{ s.t. } \forall n, y \neq Y_n : \mathbb{E}[\boldsymbol{r}^T \Delta f_n(y)] \geq 1 - \xi_n; \xi_n \geq 0. \quad (1)$$

Here, $\Delta f_n(y) = f(Y_n, \bar{\boldsymbol{z}}_n) - f(y, \bar{\boldsymbol{z}}_n)$ and $\{\xi_n\}_{n=1}^N$ are the slack variables, and $f(y, \bar{\boldsymbol{z}}_n)$ is a feature vector whose components from $(y-1)K+1$ to yK are those of the vector $\bar{\boldsymbol{z}}_n$ and all the others are 0. $\mathbb{E}[\boldsymbol{r}^T \Delta f_n(y)]$ is the "expected margin" over which the true label Y_n is preferred over a prediction y. From this viewpoint, DSLDA projects the documents onto a combined topic space and then uses a max-margin approach to predict the class label. The parameter C penalizes the margin violation of the training data.We skip the update equations here and refer the reader to [5] instead.

4 Active Doubly Supervised Latent Dirichlet Allocation (Act-DSLDA)

In the active learning setting, the model has to be changed slightly. We first state the notations used here. Suppose we are given an initial training corpus $\mathcal{L}$ with N documents belonging to Y different classes. When the learning starts, $\mathcal{L}$ is assumed to have fully labeled documents. However, as the learning progresses more documents are added to the pool $\mathcal{L}$ with class and/or a subset of supervised topics labeled. Therefore, at any intermediate point of the learning process, $\mathcal{L}$ can be assumed to contain several sets: $\mathcal{L} = \{\mathcal{T} \cup \mathcal{T}_C \cup \mathcal{T}_{A_1} \cup \mathcal{T}_{A_2} \cup \cdots \cup \mathcal{T}_{A_{K_2}}\}$, where $\mathcal{T}$ contains fully labeled documents (*i.e.* with both class and all of supervised topics labeled) and $\mathcal{T}_C$ represents the documents that have class labels. For $1 \leq k \leq K_2$, $\mathcal{T}_{A_k}$ represents the documents that have the k^{th} supervised topic labeled. Since, human provided annotations and class labels are expensive to obtain in general, we design an active learning framework where the model can query over an unlabeled pool $\mathcal{U}$ and request either class labels or a subset of the supervised topics. We use expected error reduction [18] as a measure in active selection. Such active selection mechanism is less immune to noise compared to uncertainty sampling [19] but requires the model parameters to be incrementally updated.

In the test data, the supervised topics are not observed and one has to infer them from either the parameters of the model or use some other auxiliary information. Since one of our objectives is to query over the supervised topics as well as the final category, we train a set of binary SVM classifiers that can predict the individual attributes from the features of the data. We denote the parameters of such classifiers by $\{\boldsymbol{r}_{2k}\}_{K_1<k\leq K}$. This is important to get an uncertainty measure over the supervised topics. To further clarify the issue, let us consider that only one supervised topic has to be labeled by the annotator for the n^{th} document from the set of supervised topics of size K_2. To select the most uncertain topic, one needs to compare the uncertainty of predicting the presence or absence of the individual topics. This uncertainty is different from that calculated from the conditional distribution which one might be tempted to calculate from the posterior over θ_n.

We change the notation slightly from DSLDA and denote by $\boldsymbol{r}_{1y}$ the K-dimensional real vector corresponding to the y^{th} class, and it is assumed to have a prior distribution $\mathcal{N}(0, 1/C)$. The maximization problem to generate Y_n (or the classification problem) is carried out using a max-margin principle and we use online support vector machines [20] for such update. Since the model has to be updated incrementally in the active selection step, a batch SVM solver is not applicable. Online SVM allows one to update the learnt weights incrementally given a new example.

Inference and parameter estimation in Act-DSLDA have two phases – one for the batch case when the model is trained with some labeled data and the other is for the active selection step where the model has to be incrementally updated to observe the effect of any labeled information that is queried from the oracle. In the batch mode, Act-DSLDA reduces to solving the

[1] Please see [10, 5] for further details.

following optimization problem:

$$\min_{q,\boldsymbol{\kappa}_0,\{\xi_n\}} \frac{1}{2}||\boldsymbol{r}_1||^2 - \mathcal{L}(q(\boldsymbol{Z})) + C\sum_{n=1}^{N}\xi_n \mathbb{I}_{\mathcal{T}_C,n},\ \text{s.t. } \forall n \in \mathcal{T}_C, y \neq Y_n : \mathbb{E}[\boldsymbol{r}_1^T \Delta f_n(y)] \geq 1 - \xi_n; \xi_n \geq 0. \quad (2)$$

The only difference in this objective from that of DSLDA is the presence of the indicator variable $\mathbb{I}_{\mathcal{T}_C,n}$ which is unity if the n^{th} document has a class label (*i.e.* $n \in \mathcal{T}_C$) and 0 otherwise. This implies that only the documents which have class labels are used for updating the parameters of online SVM. Rest of the updates are similar to DSLDA.

For active selection, consider that a completely unlabeled or partially labeled document, indexed by n', is to be included in the labeled pool with one of the (K_2+1) labels (one for the class label and each different supervised topic), indexed by k'. In the E step, variational parameters corresponding to all other documents except for the n'th one is kept fixed and the variational parameters for only the n'th document are updated. In the M-step, we keep the priors $\{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2\}$ over the topics and the SVM parameters $\boldsymbol{r}_2$ fixed as there is no easy way to update such parameters incrementally. From the empirical point of view, these parameters do not change much w.r.t. the variational parameters or features of a single document. However, the update of the parameters $\{\boldsymbol{\beta}, \boldsymbol{r}_1\}$ is easier. Updating $\boldsymbol{\beta}$ is accomplished by a simple update of the sufficient statistics. Updating $\boldsymbol{r}_1$ is done using the "ProcessNew" operation of online SVM followed by a few iterations of "ProcessOld".

5 Experimental Evaluation

Our evaluation used two datasets, a text corpus consisting of abstracts from ACM conferences and a multi-class image database named aYahoo [3]. Please see [5] for more details about these datasets. In order to demonstrate the contribution of each aspect of the overall model, DSLDA is compared against the following simplified models – 1. MedLDA with **o**ne-**v**s-**a**ll classification (MedLDA-OVA), 2. MedLDA with **m**ulti**t**ask **l**earning (MedLDA-MTL), 3. DSLDA with **o**nly **s**hared **s**upervised **t**opics (DSLDA-OSST), 4. DSLDA with **n**o **s**hared **l**atent **t**opics (DSLDA-NSLT), 5. **M**ajority **c**lass **m**ethod (MCM). We skip the rationales for using such baselines here to save space and details are available in [5].

Figs. 1 and 2 present representative learning curves for the image and text data respectively, showing how classification accuracy improves as the amount of class supervision (p_2) is increased. Results are shown for two different amounts of topic supervision ($p_1 = 0.5$ and $p_1 = 0.7$). The error bars in the curves show standard deviations across the 50 trials.

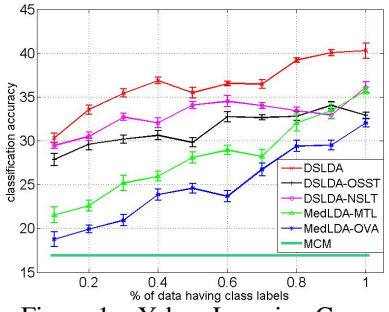

Figure 1: aYahoo Learning Curves

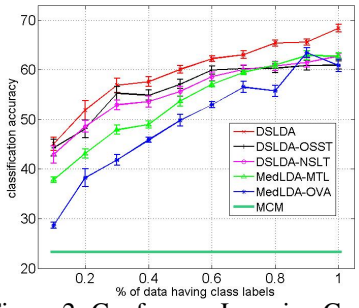

Figure 2: Conference Learning Curves

We also compare Act-DSLDA against the following models: 1. Active Learning in MedLDA with **o**ne-**v**s-**a**ll classification (Act-MedLDA-OVA) – a separate MedLDA model is trained for each class using a one-vs-all approach leaving no possibility of transfer across classes; 2. Active Learning in MedLDA with **m**ulti**t**ask **l**earning (Act-MedLDA-MTL)– a single MedLDA model is learned for all classes where the latent topics are shared across classes (this baseline is supposed to be stronger than baseline 1 where the latent topics are not shared); 3. Act-DSLDA with **o**nly **s**hared **s**upervised **t**opics (Act-DSLDA-OSST)– a model in which supervised topics are used and shared across classes but there are no latent topics (both the supervised topics and the class labels are queried using active selection strategy); 4. Act-DSLDA with **n**o **s**hared **l**atent **t**opics (Act-DSLDA-NSLT) – a model in which only supervised topics are shared across classes and a separate set of latent topics is maintained for each class (both the supervised topics and the class labels are queried using active selection strategy); 5. **R**andom **s**election of only **c**lass labels (RSC) – a MedLDA-MTL model where only the class labels are selected at random but the supervised topics are not used at all [2]; (this baseline shows the utility of active selection of classes in MedLDA-MTL framework); 6. **R**andom **s**election of **c**lass and **a**ttribute labels (RSCA) – a DSLDA model where both the class and the supervised topics are selected at random (this baseline is weaker than RSC since the supervised topics are less informative compared to the class labels).

For experiments with both image and text data in Act-DSLDA, we start with a completely labeled dataset $\mathcal{L}$ consisting of 300 documents. In every active iteration, we query for 50 labels (class labels or supervised topics). Figs. 3 and 4 present representative learning curves for the image and the text data respectively, showing how classification accuracy improves as the amount of supervision is increased. The error bars in the curves show standard deviations across the 20 trials.

[2] Note that designing a DSLDA based model where only class labels are selected at random is tricky as one needs to balance the number of supervised topics queried and the number of class labels selected at random.

Overall, the results support the hypothesis that DSLDA's ability to incorporate both supervised and latent topics allow it to achieve better predictive performance compared to baselines that exploit only one, the other, or neither. Similarly, Act-DSLDA quite consistently outperforms all of the baselines, clearly demonstrating the advantage of combining both types of topics and integrating active learning and transfer learning in the same framework.

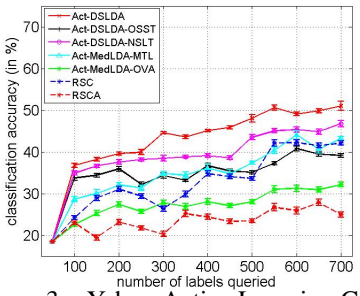

Figure 3: aYahoo Active Learning Curves

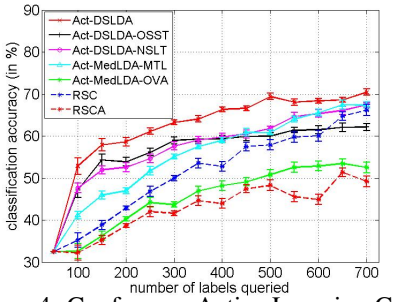

Figure 4: Conference Active Learning Curves

References

[1] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proc. of CVPR*, pages 248–255, 2009.

[2] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: A database and web-based tool for image annotation, 2008.

[3] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *Proc. of CVPR*, pages 1778–1785, 2009.

[4] A. Kovashka, S. Vijayanarasimhan, and K. Grauman. Actively selecting annotations among objects and attributes. In *Proc. of ICCV*, pages 1403–1410, 2011.

[5] A. Acharya, A. Rawal, R. J. Mooney, and E. R. Hruschka. Using both supervised and latent shared topics for multitask learning. In *ECML PKDD, Part II, LNAI 8189*, pages 369–384, 2013.

[6] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22:1345–1359, 2010.

[7] A.J. Joshi, F. Porikli, and N. Papanikolopoulos. Multi-class active learning for image classification. In *Proc. of CVPR*, pages 2372–2379, 2009.

[8] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora. In *Proc. of EMNLP*, pages 248–256, 2009.

[9] D. M. Blei and J. D. Mcauliffe. Supervised topic models. In *Proc. of NIPS*, 2007.

[10] J. Zhu, A. Ahmed, and E. P. Xing. MedLDA: maximum margin supervised topic models for regression and classification. In *Proc. of ICML*, pages 1257–1264, 2009.

[11] J. Chang and D. Blei. Relational topic models for document networks. In *Proc. of AISTATS*, 2009.

[12] R. Caruana. Multitask learning. *Machine Learning*, 28:41–75, July 1997.

[13] K. D. Bollacker and J. Ghosh. Knowledge transfer mechanisms for characterizing image datasets. In *Soft Computing and Image Processing*. Physica-Verlag, Heidelberg, 2000.

[14] A. Passos, P. Rai, J. Wainer, and H. Daumé III. Flexible modeling of latent task structures in multitask learning. In *Proc. of ICML*, pages 1103–1110, 2012.

[15] P. Rai, A. Saha, H. Daumé, III, and S. Venkatasubramanian. Domain adaptation meets active learning. In *Proc. of NAACL HLT Workshop on Active Learning for Natural Language Processing*, pages 27–32, 2010.

[16] A. Harpale and Y. Yang. Active learning for multi-task adaptive filtering. In *Proc. of ICML*, pages 431–438. Omnipress, 2010.

[17] A. Saha, P. Rai, H. Daum III, and S. Venkatasubramanian. Online learning of multiple tasks and their relationships. *JMLR - Proceedings Track*, 15:643–651, 2011.

[18] N. Roy and A. K. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proc. of ICML*, pages 441–448, 2001.

[19] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[20] A. Bordes, L. Bottou, P. Gallinari, and J. Weston. Solving multiclass support vector machines with larank. In *Proc. of ICML*, pages 89–96, 2007.

Active Multitask Learning Using Both Latent and Supervised Shared Topics

Ayan Acharya* Raymond J. Mooney* Joydeep Ghosh*

Abstract

Multitask learning (MTL) *via* shared representation has been adopted to alleviate problems with sparsity of labeled data across different learning tasks. Active learning, on the other hand, reduces the cost of labeling examples by making informative queries over the unlabeled pool of data. Therefore, a unification of both of these approaches can potentially be useful in settings where labeled information is expensive to obtain but the learning tasks or domains have some common characteristics. This paper introduces two such models – Active Doubly Supervised Latent Dirichlet Allocation (Act-DSLDA) and its non-parametric variation (Act-NPDSLDA) that integrate MTL and active learning in the same framework. These models make use of both shared latent and *supervised* topics to accomplish multitask learning. Experimental results on both document and image classification show that integrating MTL and active learning along with shared latent and supervised topics is superior to other methods which do not use all of these components.

Keywords

Active Learning, Multitask Learning, Topic Model.

1 Introduction

Designing an automated object detector in computer vision is often challenging. Many object categories abound in nature and it is expensive to get labeled examples for all of these categories. Computer vision researchers have tried to overcome such challenges by either gathering large datasets of web images [12, 15, 36] and creating benchmark datasets [1] or by formulating new algorithms that can reduce the degree of human intervention in the learning process.

One such mechanism, partly inspired by human perception and learning from high-level object descriptions, consists of learning object categories by utilizing *shared* attributes, abstract descriptors of object properties [14, 23, 22, 2]. The attributes here serve as an intermediate layer in a classifier cascade. If the *shared* attributes transcend object class boundaries, such a classifier cascade is beneficial for *transfer learning* [27] where fewer labeled examples are available for some object categories compared to others [23]. Another group of researchers have formulated methods based on active learning for reducing the expense of human annotations where the system can request labels for the most informative examples [29, 17, 19, 22].

In this paper, our objective is to combine these two orthogonal approaches in order to leverage the benefits of both – learning from a shared feature space and making active queries. In particular, we build on a recent approach proposed in [2] where multitask learning (MTL) [7] is accomplished using both shared attributes and a shared latent set of features. MTL is a form of transfer learning in which simultaneously learning multiple related "tasks" allows each one to benefit from the learning of all of the others. This approach is in contrast to "isolated" training of tasks where each task is learned independently using a separate model.

The paper is organized as follows. We present related work in Section 2, followed by the descriptions of two of our models Active Doubly Supervised Latent Dirichlet Allocation (Act-DSLDA) and a non-parametric variation of the same (Act-NPDSLDA) in Section 3 and Section 4 respectively. Experimental results on both multi-class image and document categorization are presented in Section 5. Finally, future directions and conclusions are presented in Section 6.

Note on Notation: Vectors and matrices are denoted by bold-faced lowercase and capital letters, respectively. Scalar variables are written in italic font, and sets are denoted by calligraphic uppercase letters. Dir(), Beta() and multinomial() stand for Dirichlet, Beta and multinomial distribution respectively. We also define an indicator variable $\mathbb{I}_{\mathcal{S},n} = 1$ if $n \in \mathcal{S}$ and $\mathbb{I}_{\mathcal{S},n} = 0$ if $n \notin \mathcal{S}$.

2 Background and Related Work

2.1 Statistical Topic Models LDA [4] treats documents as a mixture of topics, which in turn are defined by a distribution over a set of words. The words in a document are assumed to be sampled from multiple topics. The unsupervised LDA has been extended to account for supervision by labeling each document with its set of topics [32, 35]. In *Labeled LDA* (LLDA

*University of Texas at Austin, Austin, TX, USA. Email: {aacharya@, mooney@cs, ghosh@ece}.utexas.edu

[32]), the primary objective is to build a model of the words that indicate the presence of certain topic labels. Some other researchers [3, 44, 10] assume that supervision is provided for a single *response variable* to be predicted for a given document. In *Maximum Entropy Discriminative LDA* (MedLDA) [44], the objective is to infer some low-dimensional (topic-based) representation of documents which is predictive of the response variable. Essentially, MedLDA solves two problems jointly – dimensionality reduction and max-margin classification using the features in the dimensionally-reduced space.

2.2 Active Learning *via* Expected Error Reduction Of the several measures for selecting labels in active learning algorithms, a decision-theoretic approach called Expected Error Reduction [34] has been used quite extensively in practice [22, 38]. This approach aims to measure how much the generalization error of a model is likely to be reduced based on some labeled information y of an instance $\boldsymbol{x}$ taken from the unlabeled pool $\mathcal{U}$. The idea is to estimate the expected future error of a model trained using $\mathcal{L} \cup \langle \boldsymbol{x}, y \rangle$ on the remaining unlabeled instances in $\mathcal{U}$, and query the instance with minimal expected future error. Here $\mathcal{L}$ denotes the labeled pool of data. One approach is to minimize the expected 0/1 loss:

(2.1)
$$\boldsymbol{x}^*_{0/1} = \underset{\boldsymbol{x}}{\operatorname{argmax}} \sum_n P_{\boldsymbol{\kappa}}(y_n|\boldsymbol{x}) \left(\sum_{u=1}^{U} 1 - P_{\boldsymbol{\kappa}^{+\langle \boldsymbol{x}, y_n \rangle}} \left(\hat{y}, \boldsymbol{x}^{(u)} \right) \right).$$

where $\boldsymbol{\kappa}^{+\langle \boldsymbol{x}, y_n \rangle}$ refers to the new model after it has been re-trained with the training set $\mathcal{L} \cup \langle \boldsymbol{x}, y_n \rangle$. Note that we do not know the true label for each query instance, so we approximate using expectation over all possible labels under the current model. The objective is to reduce the expected number of incorrect predictions.

2.3 Active Knowledge Transfer There has been some effort to integrate active and transfer learning in the same framework. In [20] the authors utilized a maximum likelihood classifier to learn parameters from the source domain and use these parameters to seed the EM algorithm that explains the unlabeled data in the target domain. The example which contributed to maximum expected KL divergence of the posterior distribution with the prior distribution was selected in the active step. In [31], the source data was first used to train a classifier, the parameter of which was later updated in an online manner with new examples selected in the active step. The active selection criterion is based on uncertainty sampling [38]. Similarly, in [9], a naïve Bayes classifier is first trained with examples from the source domain and then incrementally updated with the data from the target domain selected using uncertainty sampling. In [39], the authors proposed to maintain a classifier trained from source domain(s) and the prediction of the classifier is trusted only when the likelihood of the data in the target domain is sufficiently high. In case of a lower likelihood value, domain experts are asked to label the example. The proposed method is independent of the active selection approach adopted. Harpale & Young [16] proposed active multitask learning for adaptive filtering [33] where the underlying classifier is logistic regression with Dirichlet process priors. Any feedback provided in the active selection phase improves both the task-specific and the global performance *via* a measure called *utility gain* [16]. Saha *et al.* [37] formulated an online active multitask learning framework where the information provided for one task is utilized for other tasks through a task correlation matrix. The updates are similar to perceptron updates. For active selection, they use a margin based sampling scheme which is a modified version of the sampling scheme used in [8].

Our work builds on top of a topic model framework and uses expected error reduction as active selection mechanism. Such active selection mechanism necessitates incremental update of the model parameters and hence the inference and estimation problems become challenging. This active selection mechanism is less immune to noisy observations compared to a simpler selection mechanism named uncertainty sampling [38]. Additionally, the models proposed query over both class labels and supervised topics which has not been explored in the context of MTL before.

2.4 Multitask Learning Using Both Shared Latent and Supervised Topics In multitask learning (MTL [7]), a single model is simultaneously trained to perform multiple related tasks. Many different MTL approaches have been proposed over the past 15 years (*e.g.*, see [41, 27, 28] and references therein). These include different learning methods, such as empirical risk minimization using group-sparse regularizers [21, 18], hierarchical Bayesian models [43, 24] and hidden conditional random fields [30]. In an MTL framework, if the tasks are related, training one task should provide helpful "inductive bias" for learning the other tasks.

In particular, Acharya *et al.* [2] proposed two models – **D**oubly **S**upervised **L**atent **D**irichlet **A**llocation (DSLDA) and its **n**on-**p**arametric counterpart (NPDSLDA) which support the prediction of multiple response variables based on a combination of "supervised" topics and latent topics. In the domain of computer vision, the supervised topics correspond to attributes provided by human experts. In both text and vision domains, the authors in [2] showed that incorporating both supervised and latent topics achieves better

predictive performance compared to baselines that exploit only one, the other, or neither. In our paper, we extend these models to include active selection. This extension is not trivial and several modifications have to be made in the inference and learning. With that objective in mind, the next two sub-sections discuss the incremental EM algorithm and the online support vector machine.

2.5 Incremental EM Algorithm The EM algorithm proposed by Dempster *et al.* [11] can be viewed as a joint maximization problem over $q(.)$, the conditional distribution of the hidden variables $\boldsymbol{Z}$ given the model parameters $\boldsymbol{\kappa}$ and the observed variables $\boldsymbol{X}$. The relevant objective function is given as follows:

$$F(q, \boldsymbol{\kappa}) = \mathbb{E}_q[\log(p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\kappa}))] + H(q), \tag{2.2}$$

where $H(q)$ is the entropy of the distribution $q(.)$. Often, $q(.)$ is restricted to a family of distributions $\mathcal{Q}$. It can be shown that if $\boldsymbol{\theta}^*$ is the maximizer of the above objective F then it also maximizes the likelihood of the observed data. In most of the models used in practice, the joint distribution is assumed to factorize over the instances implying that $p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\kappa}) = \prod_{n=1}^{N} p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\kappa})$. One can further restrict the family of distributions $\mathcal{Q}$ to maximize over in Eq. (2.2) to the factorized form: $q(\boldsymbol{Z}) = \prod_{n=1}^{N} q(\boldsymbol{z}_n|\boldsymbol{x}_n) = \prod_{n=1}^{N} q_n$.

An incremental variant of the EM algorithm that exploits such separability structure in both $p(.)$ and $q(.)$ was first proposed by Neal & Hinton [26]. Under such structure, the objective function in Eq. (2.2) decomposes over the observations $F(q, \boldsymbol{\theta}) = \sum_{n=1}^{N} F_n(q_n, \boldsymbol{\kappa})$, and the following incremental algorithm can instead be used to maximize F:

- **E step:** Choose some observation n to be updated over, set $q_{n'}^{(t)} = q_{n'}^{(t-1)}$ for $n' \neq n$ (no update) and set $q_n^{(t)} = \underset{q_n}{\operatorname{argmax}}\ F_n(q_n, \boldsymbol{\kappa}^{(t-1)})$.
- **M step:** $\boldsymbol{\kappa}^{(t)} = \underset{\boldsymbol{\kappa}}{\operatorname{argmax}}\ F(q^{(t)}, \boldsymbol{\kappa})$.

2.6 Online Support Vector Machines The online SVM proposed by Bordes *et al.* [5, 6] has three distinct modules that work in unison to provide a scalable learning mechanism. These modules are named as "ProcessNew", "ProcessOld" and "Optimize". All of these modules use a common operation called "SMOStep" and the only memory footprint is due to the support vectors and the associated gradient information. The module "ProcessNew" operates on a pattern that is not a support pattern. In such an update, one of the classes is chosen to be the label of the support pattern and the other class is chosen in such a way that it defines feasible direction with the highest gradient. It then performs an SMO step with the example and the selected classes. The module "ProcessOld" randomly picks a support pattern and chooses two classes that define the feasible direction with the highest gradient for that support pattern. "Optimize" resembles "ProcessOld" but picks two classes among those that correspond to existing support vectors.

3 Active Doubly Supervised Latent Dirichlet Allocation (Act-DSLDA)

Assume we are given an initial training corpus $\mathcal{L}$ with N documents belonging to Y different classes (where each document belongs to exactly one class and each class corresponds to a different task). Further assume that each of these training documents is also annotated with a set of K_2 different "supervised topics". The objective is to train a model using the words in a data, as well as the associated supervised topics and class labels, and then use this model to classify completely unlabeled test data for which no topics nor class labels are provided. When the learning starts, $\mathcal{L}$ is assumed to have fully labeled documents. However, as the learning progresses more documents are added to the pool $\mathcal{L}$ with class and/or a subset of supervised topics labeled. Therefore, at any intermediate point of the learning process, $\mathcal{L}$ can be assumed to contain several sets: $\mathcal{L} = \{\mathcal{T} \cup \mathcal{T}_C \cup \mathcal{T}_{A_1} \cup \mathcal{T}_{A_2} \cup \cdots \cup \mathcal{T}_{A_{K_2}}\}$, where $\mathcal{T}$ contains fully labeled documents (*i.e.* with both class and all of supervised topics labeled) and $\mathcal{T}_C$ represents the documents that have class labels. For $1 \leq k \leq K_2$, $\mathcal{T}_{A_k}$ represents the documents that have the k^{th} supervised topic labeled. Since, human provided annotations and class labels are expensive to obtain, we design an active learning framework where the model can query over an unlabeled pool $\mathcal{U}$ and request either class labels or a subset of the supervised topics.

We emphasize that the proposed frameworks support general MTL; however, the datasets, as explained in Section 5, happen to be multiclass, where each class is treated as a separate "task" (as typical in multi-class learning based on binary classifiers). But, in no way are the frameworks restricted to multiclass MTL.

The Act-DSLDA model is described as follows. For the n^{th} document, sample a topic selection probability vector $\boldsymbol{\theta}_n \sim \text{Dir}(\boldsymbol{\alpha}_n)$, where $\boldsymbol{\alpha}_n = \boldsymbol{\Lambda}_n \boldsymbol{\alpha}$ and $\boldsymbol{\alpha}$ is the parameter of a Dirichlet distribution of dimension K, which is the total number of topics. The topics are assumed to be of two types – latent and supervised, and

there are K_1 latent topics and K_2 supervised topics. Therefore, $K = (K_1 + K_2)$. Latent topics are never observed, while supervised topics are observed in the training data only but not in the test data. Henceforth, in each vector or matrix with K components, it is assumed that the first K_1 components correspond to the latent topics and the next K_2 components to the supervised topics. $\mathbf{\Lambda}_n$ is a diagonal binary matrix of dimension $K \times K$. The k^{th} diagonal entry is unity if *either* $1 \leq k \leq K_1$ *or* $K_1 < k \leq K$ and the n^{th} document is tagged with the k^{th} topic. Also, $\boldsymbol{\alpha} = (\boldsymbol{\alpha}^{(1)}, \boldsymbol{\alpha}^{(2)})$ where $\boldsymbol{\alpha}^{(1)}$ is a parameter of a Dirichlet distribution of dimension K_1 and $\boldsymbol{\alpha}^{(2)}$ is a parameter of a Dirichlet distribution of dimension K_2.

In the test data, the supervised topics are not observed and one has to infer them from either the parameters of the model or use some other auxiliary information. Since one of our objectives is to query over the supervised topics as well as the final category, we train a set of binary SVM classifiers that can predict the individual attributes from the features of the data. We denote the parameters of such classifiers by $\{\boldsymbol{r}_{2k}\}_{1 \leq k \leq K_2}$. This is important to get an uncertainty measure over the supervised topics. To further clarify the issue, let us consider that only one supervised topic has to be labeled by the annotator for the n^{th} document from the set of supervised topics of size K_2. To select the most uncertain topic, one needs to compare the uncertainty of predicting the presence or absence of the individual topics. This uncertainty is different from that calculated from the conditional distribution calculated from the posterior over θ_n.

For the m^{th} word in the n^{th} document, sample a topic $z_{nm} \sim \text{multinomial}(\boldsymbol{\theta}'_n)$, where $\boldsymbol{\theta}'_n = (1 - \epsilon)\{\boldsymbol{\theta}_{nk}\}_{k=1}^{k_1} \epsilon \{\Lambda_{n,kk}\boldsymbol{\theta}_{nk}\}_{k=1+k_1}^{K}$. This implies that the supervised topics are weighted by ϵ and the latent topics are weighted by $(1 - \epsilon)$. Sample the word $w_{nm} \sim \text{multinomial}(\boldsymbol{\beta}_{z_{nm}})$, where $\boldsymbol{\beta}_k$ is a multinomial distribution over the vocabulary of words corresponding to the k^{th} topic.

For the n^{th} document, generate $Y_n = \arg\max_y \boldsymbol{r}_{1y}^T \mathbb{E}(\bar{\mathbf{z}}_n)$ where Y_n is the class label associated with the n^{th} document, $\bar{\boldsymbol{z}}_n = \sum_{m=1}^{M_n} \mathbf{z}_{nm}/M_n$. Here, $\mathbf{z}_{nm}$ is an indicator vector of dimension K. $\boldsymbol{r}_{1y}$ is a K-dimensional real vector corresponding to the y^{th} class, and it is assumed to have a prior distribution $\mathcal{N}(0, 1/C)$. M_n is the number of words in the n^{th} document. The maximization problem to generate Y_n (or the classification problem) is carried out using a max-margin principle and we use online support vector machines [5, 6] for such update. Since the model has to be updated incrementally in the active selection step, a batch SVM solver is not applicable. Online SVM allows one to update the learnt weights incrementally given a new example. Note that predicting each class is treated as a separate task, and that the shared topics are useful for generalizing the performance of the model across classes. In particular, when all classes have few training examples, knowledge transfer between classes can occur through the shared topics.

3.1 Inference and Learning Inference and parameter estimation have two phases – one for the batch case when the model is trained with some labeled data and the other is for the active selection step where the model has to be incrementally updated to observe the effect of any labeled information that is queried from the oracle.

3.1.1 Learning in Batch Mode Let us denote the hidden variables by $\boldsymbol{Z} = \{\{z_{nm}\}, \{\boldsymbol{\theta}_n\}\}$, the observed variables by $\boldsymbol{X} = \{w_{nm}\}$ and the model parameters by $\boldsymbol{\kappa}_0$. The joint distribution of the hidden and observed variables is:

(3.3)
$$p(\mathbf{X}, \boldsymbol{Z}|\boldsymbol{\kappa}_0) = \prod_{n=1}^{N} p(\boldsymbol{\theta}_n|\boldsymbol{\alpha}_n) \prod_{m=1}^{M_n} p(z_{nm}|\boldsymbol{\theta}'_n) p(w_{nm}|\boldsymbol{\beta}_{\boldsymbol{z}_{nm}})$$

To avoid computational intractability, inference and estimation are performed using variational **EM**. The factorized approximation of the posterior distribution with hidden variables $\mathbf{Z}$ is given by:

$$q(\boldsymbol{Z}|\{\boldsymbol{\kappa}_n\}_{n=1}^{N}) = \prod_{n=1}^{N} q(\boldsymbol{\theta}_n|\boldsymbol{\gamma}_n) \prod_{m=1}^{M_n} q(z_{nm}|\boldsymbol{\phi}_{nm}), \tag{3.4}$$

where $\boldsymbol{\theta}_n \sim \text{Dir}(\boldsymbol{\gamma}_n)\ \forall n \in \{1, 2, \cdots, N\}$, $z_{nm} \sim \text{multinomial}(\boldsymbol{\phi}_{nm})\ \forall n \in \{1, 2, \cdots, N\}$ and $\forall m \in \{1, 2, \cdots, M_n\}$, and $\boldsymbol{\kappa}_n = \{\boldsymbol{\gamma}_n, \{\boldsymbol{\phi}_{nm}\}\}$, which is the set of variational parameters corresponding to the n^{th} instance. Further, $\boldsymbol{\gamma}_n = (\gamma_{nk})_{k=1}^{K}\ \forall n$, and $\boldsymbol{\phi}_{nm} = (\phi_{nmk})_{k=1}^{K}\ \forall n, m$. With the use of the lower bound obtained by the factorized approximation, followed by Jensen's inequality, Act-DSLDA reduces to solving the following optimization problem[1]:

$$\min_{q, \boldsymbol{\kappa}_0, \{\xi_n\}} \frac{1}{2}||\boldsymbol{r}_1||^2 - \mathcal{L}(q(\boldsymbol{Z})) + C \sum_{n=1}^{N} \xi_n \mathbb{I}_{\mathcal{T}_C, n},$$

$$\text{s.t. } \forall n \in \mathcal{T}_C, y \neq Y_n : \mathbb{E}[\boldsymbol{r}_1^T \Delta f_n(y)] \geq 1 - \xi_n; \xi_n \geq 0. \tag{3.5}$$

Here, $\Delta f_n(y) = f(Y_n, \bar{\boldsymbol{z}}_n) - f(y, \bar{\boldsymbol{z}}_n)$ and $\{\xi_n\}_{n=1}^{N}$ are the slack variables, and $f(y, \bar{\boldsymbol{z}}_n)$ is a feature vector

[1] Please see [44] for further details.

whose components from $(y-1)K+1$ to yK are those of the vector $\bar{\boldsymbol{z}}_n$ and all the others are 0. $\mathbb{E}[\boldsymbol{r}_1^T \Delta f_n(y)]$ is the "expected margin" over which the true label Y_n is preferred over a prediction y. From this viewpoint, Act-DSLDA projects the documents onto a combined topic space and then uses a max-margin approach to predict the class label. The parameter C penalizes the margin violation of the training data. The indicator variable $\mathbb{I}_{\mathcal{T}_C,n}$ is unity if the n^{th} document has a class label (*i.e.* $n \in \mathcal{T}_C$) and 0 otherwise. This implies that only the documents which have class labels are used for updating the parameters of online SVM.

Let $\mathcal{Q}$ be the set of all distributions having a fully factorized form as given in (3.4). Note that such factorized approximation makes the use of incremental variation of EM possible in the active selection step following the discussion in Section 2.5. Let the distribution q^* from the set $\mathcal{Q}$ optimize the objective in Eq. (3.5). The optimal values of the corresponding variational parameters are same as those of DSLDA [2]. The optimal values of $\boldsymbol{\phi}_{nm}$ depend on $\boldsymbol{\gamma_n}$ and vice-versa. Therefore, iterative optimization is adopted to maximize the lower bound until convergence is achieved.

During testing, one does not observe a document's supervised topics and instead an approximate solution, as also used in [32, 2], is employed where the variables $\{\boldsymbol{\Lambda}_n\}$ are assumed to be absent altogether in the test phase, and the problem is treated as inference in MedLDA with K latent topics.

In the M step, the objective in Eq. (3.5) is maximized w.r.t $\boldsymbol{\kappa}_0$. The optimal value of β_{kv} is again similar to that of DSLDA [2]. However, numerical methods for optimization are required to update $\boldsymbol{\alpha}_1$ or $\boldsymbol{\alpha}_2$. The update for the parameters $\{\boldsymbol{r}_{1y}\}_{y=1}^{Y}$ is carried out using online SVM [5, 6] following Eq. (3.5).

3.1.2 Incremental Learning in Active Selection

The method of Expected Entropy Reduction requires one to take a data point from the unlabeled pool and one of its possible labels, update the model, and observe the generalized error on the unlabeled pool. This process is computationally expensive unless there is an efficient way to update the model incrementally. The incremental view of EM and the online SVM framework are appropriate for such update.

Consider that a completely unlabeled or partially labeled document, indexed by n', is to be included in the labeled pool with one of the (K_2+1) labels (one for the class label and each different supervised topic), indexed by k'. In the E step, variational parameters corresponding to all other documents except for the n'th one is kept fixed and the variational parameters for only the n'th document are updated. In the M-step, we keep the priors $\{\boldsymbol{\alpha}^{(1)}, \boldsymbol{\alpha}^{(2)}\}$ over the topics and the SVM parameters $\boldsymbol{r}_2$ fixed as there is no easy way to update such parameters incrementally. From the empirical point of view, these parameters do not change much w.r.t. the variational parameters (or features in topic space representation) of a single document. However, the update of the parameters $\{\boldsymbol{\beta}, \boldsymbol{r}_1\}$ is easier. Updating $\boldsymbol{\beta}$ is accomplished by a simple update of the sufficient statistics. Updating $\boldsymbol{r}_1$ is done using the "ProcessNew" operation of online SVM followed by a few iterations of "ProcessOld". The selection of the document-label pair is guided by the measure given in Eq. (2.1). Note that since SVM uses hinge loss which, in turn, upper bounds the $0-1$ loss in classification, use of the measure from Eq. (2.1) for active query selection is justified.

From the modeling perspective, the difference between DSLDA [2] and Act-DSLDA lies in maintaining attribute classifiers and ignoring documents in the max-margin learning that do not have any class label. Online SVM for max-margin learning is essential in the batch mode just to maintain the support vectors and incrementally update them in the active selection step. One could also use incremental EM in the batch mode. However, that takes up lot of computation time when the labeled dataset gets larger, as the E step for each document is followed by an M-step in incremental EM.

4 Active Non-parametric DSLDA (Act-NPDSLDA)

A non-parametric extension of Act-DSLDA (Act-NPDSLDA) automatically determines the best number of latent topics for modeling the given data. A modified stick breaking construction of Hierarchical Dirichlet Process (HDP), recently introduced in [40] is used here which makes variational inference feasible. The Act-NPDSLDA model is presented below.

- Sample $\boldsymbol{\phi}_{k_1} \sim \text{Dir}(\boldsymbol{\eta}_1)\ \forall k_1 \in \{1, 2, \cdots, \infty\}$ and $\boldsymbol{\phi}_{k_2} \sim \text{Dir}(\boldsymbol{\eta}_2)\ \forall k_2 \in \{1, 2, \cdots, K_2\}$. $\boldsymbol{\eta}_1, \boldsymbol{\eta}_2$ are the parameters of Dirichlet distribution of dimension V. Also, sample $\beta'_{k_1} \sim \text{Beta}(1, \delta_0)\ \forall k_1 \in \{1, 2, \cdots, \infty\}$.

- For the n^{th} document, sample $\boldsymbol{\pi}_n^{(2)} \sim \text{Dir}(\boldsymbol{\Lambda}_n \boldsymbol{\alpha}^{(2)})$. $\boldsymbol{\alpha}^{(2)}$ is the parameter of Dirichlet of dimension K_2. $\boldsymbol{\Lambda}_n$ is a diagonal binary matrix of dimension $K_2 \times K_2$. The k^{th} diagonal entry is unity if the n^{th} word is tagged with the k^{th} supervised topic. Similar to the case of Act-DSLDA, in the test data, the supervised topics are not observed and set of binary SVM classifiers, trained with document-attribute pair data, are used instead that can predict the individual attributes from the features of the data. The parameters of such classifiers are denoted by $\{\boldsymbol{r}_{2k}\}_{1 \le k \le K_2}$.

- $\forall n, \forall t \in \{1, 2, \cdots, \infty\}$, sample $\pi'_{nt} \sim \text{Beta}(1, \alpha_0)$.

Assume $\boldsymbol{\pi}_n^{(1)} = (\pi_{nt})_t$ where $\pi_{nt} = \pi'_{nt}\prod_{l<t}(1-\pi'_{nl})$. $\forall n, \forall t$, sample $c_{nt} \sim$ multinomial$(\boldsymbol{\beta})$ where $\beta_{k_1} = \beta'_{k_1}\prod_{l<k_1}(1-\beta'_l)$. $\boldsymbol{\pi}_n^{(1)}$ represents the probability of selecting the sampled atoms in $\boldsymbol{c}_n$.

• For the m^{th} word in the n^{th} document, sample $z_{nm} \sim$ multinomial$((1-\epsilon)\boldsymbol{\pi}_n^{(1)}, \epsilon\boldsymbol{\pi}_n^{(2)})$. This implies that with probability ϵ, a topic is selected from the set of supervised topics and with probability $(1-\epsilon)$, a topic is chosen from the set of (infinite number of) unsupervised topics. Sample w_{nm} from a multinomial given by Eq. (3).

• For the n^{th} document, generate $Y_n = \arg\max_y \boldsymbol{r}_{1y}^T \mathbb{E}(\bar{\mathbf{z}}_n)$ where Y_n is the class label associated with the n^{th} document, $\bar{\mathbf{z}}_n = \sum_{m=1}^{M_n} \mathbf{z}_{nm}/M_n$. The maximization problem to generate Y_n (or the classification problem) is carried out using an online support vector machine. The joint distribution of the hidden and observed variables is given in Eq. (1).

4.1 Inference and Learning

4.1.1 Learning in Batch Mode As an approximation to the posterior distribution over the hidden variables, we use the factorized distribution given in Eq. (2). $\boldsymbol{\kappa}_0$ and $\boldsymbol{\kappa}$ denote the sets of model and variational parameters, respectively. $\bar{K_1}$ is the truncation limit of the corpus-level Dirichlet Process and T is the truncation limit of the document-level Dirichlet Process. $\{\boldsymbol{\lambda}_k\}$ are the parameters of Dirichlet each of dimension V. $\{u_{k_1}, v_{k_1}\}$ and $\{a_{nt}, b_{nt}\}$ are the parameters of Beta distribution corresponding to corpus level and document level sticks respectively. $\{\boldsymbol{\varphi}_{nt}\}$ are multinomial parameters of dimension $\bar{K_1}$ and $\{\boldsymbol{\zeta}_{nm}\}$ are multinomials of dimension $(T+K_2)$. $\{\boldsymbol{\gamma}_n\}_n$ are parameters of Dirichlet distribution of dimension K_2.

The underlying optimization problem takes the same form as in Eq. (3.5). The only difference lies in the calculation of $\Delta f_n(y) = f(Y_n, \bar{\boldsymbol{s}}_n) - f(y, \bar{\boldsymbol{s}}_n)$. The first set of dimensions of $\bar{\boldsymbol{s}}_n$ (corresponding to the unsupervised topics) is given by $1/M_n \sum_{m=1}^{M_n} \boldsymbol{c}_{nz_{nm}}$, where $\boldsymbol{c}_{nt}$ is an indicator vector over the set of unsupervised topics. The following K_2 dimensions (corresponding to the supervised topics) are given by $1/M_n \sum_{m=1}^{M_n} \boldsymbol{z}_{nm}$. After the variational approximation with $\bar{K_1}$ number of corpus level sticks, $\bar{\boldsymbol{s}}_n$ turns out to be of dimension $(\bar{K_1}+K_2)$ and the feature vector $f(y, \bar{\boldsymbol{s}}_n)$ constitutes $Y(\bar{K_1}+K_2)$ elements. The components of $f(y, \bar{\boldsymbol{s}}_n)$ from $(y-1)(\bar{K_1}+K_2)+1$ to $y(\bar{K_1}+K_2)$ are those of the vector $\bar{\boldsymbol{s}}_n$ and all the others are 0. The E-step update equations of Act-NPDSLDA are similar to NP-DSLDA [2]. The M-step updates are similar to Act-DSLDA and are omitted here due to space constraints.

4.1.2 Incremental Learning in Active Selection Again assume that a completely unlabeled or partially labeled document, indexed by n', is to be included in the labeled pool with the k'th label. In the E step, variational parameters corresponding to all other documents except for the n'th one is kept fixed and the variational parameters for only the n'th document are updated. The incremental update of the "global" variational parameters $\{u_{k_1}, v_{k_1}\}_{k_1=1}^{K_1}$ is also straightforward following the equations given in [2]. In the M-step, we keep the priors $\{\boldsymbol{\eta}_1, \boldsymbol{\eta}_2, \boldsymbol{\alpha}^{(2)}\}$ and the SVM parameters $\boldsymbol{r}_2$ fixed but the parameters $\boldsymbol{r}_1$ are updated using online SVM.

5 Experimental Results

5.1 Data Description Our evaluation used two datasets, a text corpus and a multi-class image database, as described below.

5.1.1 aYahoo Data The first set of experiments was conducted with the aYahoo image dataset from [14] which has 12 classes – carriage, centaur, bag, building, donkey, goat, jetski, monkey, mug, statue, wolf, and zebra. Each image is annotated with relevant visual attributes such as "has head", "has wheel", "has torso" and 61 others, which we use as the supervised topics. After extracting SIFT features [25] from the raw images, quantization into 250 clusters is performed, defining the vocabulary for a bag of visual words. Images with less than two attributes were discarded. The resulting dataset of size 2275 was equally split into training and test data.

5.1.2 ACM Conference Data The text corpus consists of conference paper abstracts from two groups of conferences. The first group has four conferences related to data mining – WWW, SIGIR, KDD, and ICML, and the second group consists of two VLSI conferences – ISPD and DAC. The classification task is to determine the conference at which the abstract was published. As supervised topics, we use keywords provided by the authors, which are presumably useful in determining the conference venue. A total of 2,300 abstracts were collected each of which had at least three keywords and an average of 78 ($\pm$33.5) words. After stop-word removal, the vocabulary size for the assembled data is 13,412 words. The number of supervised topics is 55. The resulting dataset was equally split into training and test data.

5.2 Methodology In order to demonstrate the contribution of each aspect of the overall model, Act-

Joint Distribution of Act-NPDSLDA

$$p(\mathbf{X}, \boldsymbol{Z}|\boldsymbol{\kappa}_0) = \prod_{k_1=1}^{\infty} p(\phi_{k_1}|\boldsymbol{\eta}_1)p(\beta'_{k_1}|\boldsymbol{\delta}_0) \prod_{k_2=1}^{K_2} p(\phi_{k_2}|\boldsymbol{\eta}_2) \prod_{n=1}^{N} p(\boldsymbol{\pi}_n^{(2)}|\boldsymbol{\alpha}_2) \prod_{t=1}^{\infty} p(\boldsymbol{\pi}_{nt}^{\prime(1)}|\boldsymbol{\alpha}_0)p(c_{nt}|\boldsymbol{\beta}') \prod_{m=1}^{M_n} p(z_{nm}|\boldsymbol{\pi}_n^{(1)}, \boldsymbol{\pi}_n^{(2)}, \epsilon)p(w_{nm}|\boldsymbol{\phi}, c_{nz_{nm}}, z_{nm}). \quad (1)$$

Variational Distribution of Act-NPDSLDA

$$q(\boldsymbol{Z}|\boldsymbol{\kappa}) = \prod_{k_1=1}^{\bar{K_1}} q(\phi_{k_1}|\boldsymbol{\lambda}_{k_1}) \prod_{k_2=1}^{K_2} q(\phi_{k_2}|\boldsymbol{\lambda}_{k_2}) \prod_{k_1=1}^{\bar{K}_1-1} q(\beta'_{k_1}|u_{k_1}, v_{k_1}) \prod_{n=1}^{N} q(\boldsymbol{\pi}_n^{(2)}|\boldsymbol{\gamma}_n) \prod_{t=1}^{T-1} q(\boldsymbol{\pi}_{nt}^{\prime(1)}|a_{nt}, b_{nt}) \prod_{t=1}^{T} q(c_{nt}|\boldsymbol{\varphi}_{nt}) \prod_{m=1}^{M_n} q(z_{nm}|\boldsymbol{\zeta}_{nm}). \quad (2)$$

Multinomial Distribution for Sampling Words in Act-NPDSLDA

$$\prod_{k_1=1}^{\infty} \prod_{v=1}^{V} \phi_{k_1 v}^{\mathbb{I}\{w_{nm}=v\}\mathbb{I}\{c_{nz_{nm}}=k_1 \in \{1,\cdots,\infty\}\}} \prod_{k_2=1}^{K_2} \prod_{v=1}^{V} \phi_{k_2 v}^{\mathbb{I}\{w_{nm}=v\}\mathbb{I}\{z_{nm}=k_2 \in \{1,\cdots,K_2\}\}}. \quad (3)$$

Table 1: Distributions in Act-NPDSLDA

DSLDA and Act-NPDSLDA are compared against the following simplified models[2]:

• Active Learning in MedLDA with **o**ne-**v**s-**a**ll classification (Act-MedLDA-OVA): A separate MedLDA model is trained for each class using a one-vs-all approach leaving no possibility of transfer across classes. Supervised topics are not included in such modeling and the class labels are also obtained using active learning.

• Active Learning in MedLDA with **m**ulti**t**ask **l**earning (Act-MedLDA-MTL): A single MedLDA model is learned for all classes where the latent topics are shared across classes. Again, supervised topics are not used and the class labels are obtained using active learning. This baseline is supposed to be stronger than baseline 1 where the latent topics are not shared.

• Act-DSLDA with **o**nly **s**hared **s**upervised **t**opics (Act-DSLDA-OSST): A model in which supervised topics are used and shared across classes but there are no latent topics. Both the supervised topics and the class labels are queried using active selection strategy.

• Act-DSLDA with **n**o **s**hared **l**atent **t**opics (Act-DSLDA-NSLT): A model in which only supervised topics are shared across classes and a separate set of latent topics is maintained for each class. Both the supervised topics and the class labels are queried using active selection strategy. This model has stronger representation capability compared to Act-DSLDA-OSST which do not use any latent topics at all.

• **R**andom **s**election of only **c**lass labels (RSC) – A MedLDA-MTL model where only the class labels are selected at random but the supervised topics are not used at all. Note that designing a DSLDA based model where only class labels are selected at random is tricky as one needs to balance the number of supervised topics queried and the number of class labels selected at random. This baseline shows the utility of active selection of classes in MedLDA-MTL framework.

[2]Note that MedLDA is a state-of-the-art supervised topic model.

• **R**andom **s**election of **c**lass and **a**ttribute labels (RSCA) – A DSLDA model where both the class and the supervised topics are selected at random. This baseline is weaker than RSC since the supervised topics are less informative compared to the class labels. Both of RSC and RSCA are used to exhibit the utility of active learning for class and supervised topic selection.

5.3 Results In the experiments with both image and text data, we start with a completely labeled dataset $\mathcal{L}$ consisting of 300 documents. In every active iteration, we query for 50 labels (class labels or supervised topics). Figs. 1 and 2 present representative learning curves for the image and the text data respectively, showing how classification accuracy improves as the amount of supervision is increased. The error bars in the curves show standard deviations across the 20 trials.

5.4 Discussion The results demonstrate that Act-DSLDA and Act-NPDSLDA quite consistently outperform all of the baselines, clearly demonstrating the advantage of combining both types of topics and integrating active learning and transfer learning in the same framework. Act-NPDSLDA performs about as well or better as Act-DSLDA, for which the optimal number of latent topics has been chosen using an expensive model-selection search.

As to be expected, the active DSLDA methods' advantage over their random selection method (RSC) is greatest at the lower end of the learning curve. Act-MedLDA-OVA does little better than RSCA showing that the active selection of class labels helps even if there is no transfer across classes. Act-MedLDA-MTL consistently outperforms Act-MedLDA-OVA as well as RSC showing that active transfer learning is beneficial for MedLDA-MTL. Act-DSLDA-OSST does better than both Act-MedLDA-MTL and RSC towards the lower end of the learning curve but with more labeled information this model does not perform that well since it does not use latent topics. Act-DSLDA-

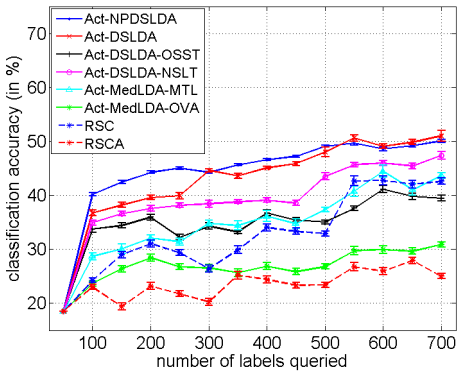

Figure 1: aYahoo Learning Curves

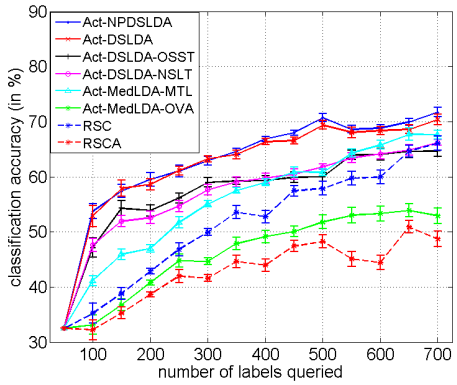

Figure 2: ACM Conference Learning Curves

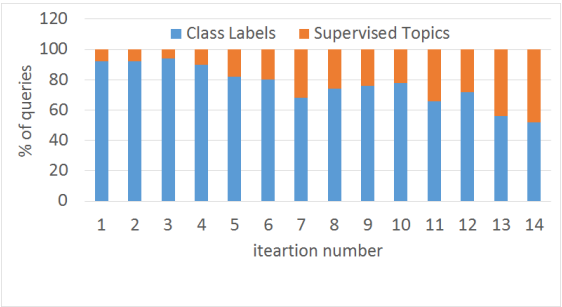

Figure 3: aYahoo Query Distribution

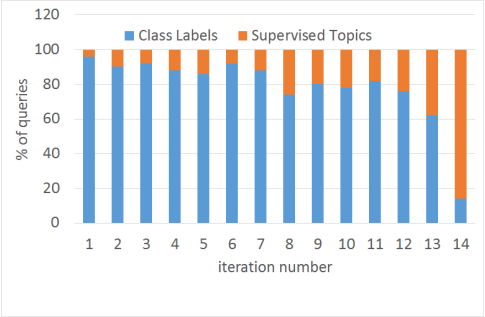

Figure 4: ACM Conference Query Distribution

NSLT also performs better than Act-DSLDA-OSST because the former utilizes latent topics.

Figs. 3 and 4 show the percentage (out of 50 queries) of class labels and supervised topics queried by Act-DSLDA in vision and text data respectively at each iteration step. Initially, the model queries for more class labels but towards the end of the learning curve, more supervised topics are queried. By the 14^{th} iteration, the class labels of all the documents in the training set are queried. From the 15^{th} iteration onwards, only supervised topics are queried. This observation is not that surprising since the class labels are more discriminative compared to the supervised topics and hence are queried more. However, queries of supervised topics are also helpful and allow continued improvement later in the learning curve (not that significant though).

6 Future Work and Conclusion

This paper has introduced two new models for active multitask learning. Experimental results comparing to six different ablations of these models demonstrate the utility of integrating active and multitask learning in one framework that also unifies latent and supervised shared topics. One could additionally actively query for rationales [42, 13] and further improve the predictive performance.

References

[1] *The PASCAL visual object classes (VOC) challenge*, International Journal of Computer Vision, 88 (2010), pp. 303–338.

[2] A. Acharya, A. Rawal, R. J. Mooney, and E. R. Hruschka, *Using both supervised and latent shared topics for multitask learning*, in ECML PKDD, Part II, LNAI 8189, 2013, pp. 369–384.

[3] D. M. Blei and J. D. Mcauliffe, *Supervised topic models*, in Proc. of NIPS, 2007.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan, *Latent Dirichlet Allocation*, JMLR, 3 (2003), pp. 993–1022.

[5] A. Bordes, L. Bottou, P. Gallinari, and J. Weston, *Solving multiclass support vector machines with larank*, in Proc. of ICML, 2007, pp. 89–96.

[6] A. Bordes, S. Ertekin, J. Weston, and L. Bottou, *Fast kernel classifiers with online and active learning*, JMLR, 6 (2005), pp. 1579–1619.

[7] R. Caruana, *Multitask learning*, Machine Learning, 28 (1997), pp. 41–75.

[8] Nicolò Cesa-Bianchi, Claudio Gentile, and Luca Zaniboni, *Worst-case analysis of selective sampling for linear classification*, JMLR, 7 (2006), pp. 1205–1230.

[9] Y. S. Chan and H. T. Ng, *Domain adaptation with active learning for word sense disambiguation*, in Proc. of ACL, 2007, pp. 49–56.

[10] J. Chang and D. Blei, *Relational topic models for document networks*, in Proc. of AISTATS, 2009.

[11] A. P. Dempster, N. M. Laird, and D. B. Rubin, *Maximum likelihood from incomplete data via the EM algorithm*, J. Royal Statistical Society. Series B (Methodological), 39 (1977), pp. 1–38.

[12] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, *ImageNet: A large-scale hierarchical image database*, in Proc. of CVPR, 2009, pp. 248–255.

[13] J. Donahue and K. Grauman, *Annotator rationales for visual recognition*, in Proc. of ICCV, 2011, pp. 1395–1402.

[14] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, *Describing objects by their attributes*, in Proc. of CVPR, 2009, pp. 1778–1785.

[15] V. Ferrari and A. Zisserman, *Learning visual attributes*, in Proc. of NIPS, 2007.

[16] Abhay Harpale and Yiming Yang, *Active learning for multi-task adaptive filtering*, in Proc. of ICML, Omnipress, 2010, pp. 431–438.

[17] P. Jain and A. Kapoor, *Active learning for large multi-class problems*, in Proc. of CVPR, 2009, pp. 762–769.

[18] R. Jenatton, J. Audibert, and F. Bach, *Structured variable selection with sparsity-inducing norms*, JMLR, 12 (2011), pp. 2777–2824.

[19] A.J. Joshi, F. Porikli, and N. Papanikolopoulos, *Multi-class active learning for image classification*, in Proc. of CVPR, 2009, pp. 2372–2379.

[20] G. Jun and J. Ghosh, *An efficient active learning algorithm with knowledge transfer for hyperspectral remote sensing data*, in Proc. of International Geosci. and Sens. Symposium, vol. 1, 2008, pp. I–52–I–55.

[21] S. Kim and E. P. Xing, *Tree-guided group lasso for multi-task regression with structured sparsity*, in Proc. of ICML, 2010, pp. 543–550.

[22] A. Kovashka, S. Vijayanarasimhan, and K. Grauman, *Actively selecting annotations among objects and attributes*, in Proc. of ICCV, 2011, pp. 1403–1410.

[23] C. H. Lampert, H. Nickisch, and S. Harmeling, *Learning to detect unseen object classes by betweenclass attribute transfer*, in Proc. of CVPR, 2009, pp. 951–958.

[24] Y. Low, D. Agarwal, and A. J. Smola, *Multiple domain user personalization*, in Proc. of KDD, 2011, pp. 123–131.

[25] D. G. Lowe, *Distinctive image features from scale-invariant keypoints*, International Journal of Computer Vision, 60 (2004), pp. 91–110.

[26] R. M. Neal and G. E. Hinton, *A view of the EM algorithm that justifies incremental, sparse, and other variants*, 1999.

[27] S. J. Pan and Q. Yang, *A survey on transfer learning*, IEEE Transactions on Knowledge and Data Engineering, 22 (2010), pp. 1345–1359.

[28] A. Passos, P. Rai, J. Wainer, and H. Daumé III, *Flexible modeling of latent task structures in multitask learning*, in Proc. of ICML, 2012, pp. 1103–1110.

[29] G.J. Qi, Xian-Sheng H., Yong R., Jinhui T., and Hong-Jiang Z., *Two-dimensional active learning for image classification*, in Proc. of CVPR, 2008, pp. 1–8.

[30] A. Quattoni, S. Wang, L. P Morency, M. Collins, and T. Darrell, *Hidden-state conditional random fields*, in IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007.

[31] P. Rai, A. Saha, H. Daumé, III, and S. Venkatasubramanian, *Domain adaptation meets active learning*, in Proc. of NAACL HLT Workshop on Active Learning for Natural Language Processing, 2010, pp. 27–32.

[32] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning, *Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora*, in Proc. of EMNLP, 2009, pp. 248–256.

[33] Stephen Robertson and Ian Soboroff, *The trec 2002 filtering track report*, in Text Retrieval Conference, 2002.

[34] N. Roy and A. K. McCallum, *Toward optimal active learning through sampling estimation of error reduction*, in Proc. of ICML, 2001, pp. 441–448.

[35] T. N. Rubin, A. Chambers, P. Smyth, and M. Steyvers, *Statistical topic models for multi-label document classification*, CoRR, abs/1107.2462, 2011.

[36] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, *Labelme: A database and web-based tool for image annotation*, 2008.

[37] Avishek Saha, Piyush Rai, Hal Daum III, and Suresh Venkatasubramanian, *Online learning of multiple tasks and their relationships.*, JMLR - Proceedings Track, 15 (2011), pp. 643–651.

[38] B. Settles, *Active learning literature survey*, Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[39] X. Shi, W. Fan, and J. Ren, *Actively transfer domain knowledge*, in Proc. of ECML PKDD - Part II, 2008, pp. 342–357.

[40] C. Wang, J. W. Paisley, and D. M. Blei, *Online variational inference for the hierarchical Dirichlet process*, JMLR - Proceedings Track, 15 (2011), pp. 752–760.

[41] K. Weinberger, A. Dasgupta, J. Langford, A. Smola, and J. Attenberg, *Feature hashing for large scale multitask learning*, in Proc. of ICML, 2009, pp. 1113–1120.

[42] O. F. Zaidan, J. Eisner, and C. Piatko, *Machine learning with annotator rationales to reduce annotation cost*, in Proc. of the NIPS Workshop on Cost Sensitive Learning, 2008.

[43] J. Zhang, Z. Ghahramani, and Y. Yang, *Flexible latent variable models for multi-task learning*, Machine Learning, 73 (2008), pp. 221–242.

[44] J. Zhu, A. Ahmed, and E. P. Xing, *MedLDA: maximum margin supervised topic models for regression and classification*, in Proc. of ICML, 2009, pp. 1257–1264.