INFERENCE IN SUPERVISED LATENT DIRICHLET ALLOCATION

Balaji Lakshminarayanan[1*] *and Raviv Raich*[2]

[1] Yandex Labs, 299 S California Ave, Suite 200, Palo Alto, CA 94306
[2] School of EECS, Oregon State University, Corvallis, OR 97331
balaji@yandex-team.ru, raich@eecs.oregonstate.edu

ABSTRACT

Supervised latent Dirichlet allocation (Supervised-LDA) [1] is a probabilistic topic model that can be used for classification. One of the advantages of Supervised-LDA over unsupervised LDA is that it can potentially learn topics that are inline with the class label. The variational Bayes algorithm proposed in [1] for inference in Supervised-LDA suffers from high computational complexity. To address this issue, we develop computationally efficient inference methods for Supervised-LDA. Specifically, we present collapsed variational Bayes and MAP inference for parameter estimation in Supervised-LDA. Additionally, we present computationally efficient inference methods to determine the label of unlabeled data. We provide an empirical evaluation of the classification performance and computational complexity (training as well as classification runtime) of different inference methods for the Supervised-LDA model and a classifier based on probabilistic latent semantic analysis.

Index Terms— Supervised Latent Dirichlet Allocation, Bayesian inference, Classification

1. INTRODUCTION

Latent Dirichlet allocation (LDA) [2] is an unsupervised latent variable model originally applied in the field of document modeling due to its ability to decompose documents into topics and uncover topics decomposition into words in a concise manner. As an unsupervised model, LDA can be used to perform dimensionality reduction by mapping the high dimensional bag-of-words representation to lower dimensional topic representation.

Recently, there has been a growing interest in supervised extensions of LDA for applications such as image classification [1, 3, 4], document classification [5–7], movie rating prediction [8], named entity mining [7, 9], and credit attribution in multi-labeled corpora [10]. In this paper, we focus on the supervised LDA model introduced in [1] (henceforth referred to as Supervised-LDA). The motivations for supervised topic models are multi fold. Supervised topic models can help in identifying topics specific to a particular class. In addition, probabilistic models are flexible, allowing simultaneous modeling of various types of information, for instance, the Supervised-LDA model can be readily extended to handle multiple-labels and additional information such as annotations or tags.

Despite the ability of topic models to produce a concise representation, parameter estimation in topic models remains a challenging task. In most cases, exact inference is intractable and hence, approximate inference methods are required. Inference methods for topic models can be broadly categorized into sampling based approaches and deterministic approximations. Recent work stresses the importance of properly adapting the priors (hyperparameters) in LDA-based models [11, 12]. This can be addressed by optimizing the prior parameterization instead of using a fixed prior, a task which increases the computational complexity associated with inference in LDA-based models. Understanding the tradeoff between computational complexity and classification performance of the different inference methods for Supervised-LDA is key to identifying the most suitable inference algorithm for a particular application. While we present results only for the Supervised-LDA model, we believe that similar trends will hold for other labeled/discriminative topic models as well. For an excellent comparison of different inference methods such as variational Bayes (VB), collapsed Gibbs sampling (CGS), collapsed variational Bayes (CVB) and maximum a-posteriori (MAP) inference for (unsupervised) LDA, we refer to [12].

Previous work in the Supervised-LDA model employed VB for inference [1]. We derive the MAP and CVB inference solutions for Supervised-LDA and study the effect of the choice of inference method for Supervised-LDA. While the extension of [12] to the supervised case might appear straightforward at first sight, several new aspects arise in the supervised case:

- *Model based classification:* The classification stage is completely new relative to LDA and requires the development of efficient inference techniques for classification of test documents.
- *Classification accuracy:* Supervised-LDA is evaluated in terms of classification accuracy rather than perplexity. It is not obvious which inference method leads to the best classification accuracy.
- *Train vs Test computational complexity:* While the training complexity of Supervised-LDA is similar to that of LDA, model based classification approach for Supervised-LDA requires significant additional computation in the test stage than LDA, and can be computationally intensive when the number of classes is large. We introduce a new classification approach to solve this problem.

In this work, we address the following question: Which inference method provides a good trade-off between classification accuracy and computational complexity for the Supervised-LDA model?

2. PROBLEM STATEMENT

The training data is assumed to be a collection of M documents along with their corresponding labels. The collection of N_i words for the i^{th} document is denoted by $\mathbf{w}_i = \{w_{i1}, \ldots, w_{iN_i}\}$ and the label associated with the i^{th} document is denoted by y_i. The entire corpus can then be represented by $(\mathbf{W}, \mathbf{Y})$ where

[*] The first author performed this work while at Oregon State University.

Table 1. List of symbols

C	Number of classes
M	Number of training documents
N_i	Number of words in i^{th} training document
K	Number of topics
V	Vocabulary size
$\mathbf{W}$	All the words in the training documents
$\mathbf{w}_i$	$1 \times N_i$ vector containing the words in document i
$\mathbf{z}_i$	$1 \times N_i$ vector containing topic assignments of corresponding words in $\mathbf{w}_i$
$\mathbf{Y}$	Labels of the training documents
$\boldsymbol{\theta}$	$K \times M$ matrix whose i^{th} column represents the 'topic-multinomial' parameter for the i^{th} training document
$\boldsymbol{\phi}$	$K \times V$ matrix where ϕ_{kv} denotes the probability of v^{th} word given k^{th} topic
$\boldsymbol{\alpha}$	$C \times K$ matrix where $\alpha_{c,\cdot}$ denotes the Dirichlet prior for class c
$\boldsymbol{\beta}$	$1 \times V$ vector which denotes the Dirichlet prior for each row of $\boldsymbol{\phi}$

$\mathbf{Y} = (y_1, y_2, \ldots, y_M)$ and $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_M)$. We assume that each document belongs to one of C classes, i.e., $y_i \in \{1, 2, \ldots, C\}$. We refer the reader to Table 1 for an explanation of the symbols used. The task is to learn a model for $(\mathbf{W}, \mathbf{Y})$ so that we are able to classify a new test document $\mathbf{w}_t$. Next, we discuss the details of the Supervised-LDA model used in this paper.

3. DESCRIPTION OF THE MODEL – SUPERVISED LDA

Supervised-LDA [1] is a natural extension to the original LDA model [2]. The graphical model for Supervised-LDA is shown in Fig. 1 and the generative process is explained in Algorithm 1.

Algorithm 1 Generative process

for $k = 1$ to K **do**
  Draw $\phi_{k,\cdot} \sim$ Dirichlet($\boldsymbol{\beta}$)
end for
for $i = 1$ to M **do**
  Draw $y_i \sim P(y), N_i$
  Draw $\theta_i \sim$ Dirichlet($\alpha_{y_i,\cdot}$)
  for $j = 1$ to N_i **do**
    Draw $z_{ij} \sim$ Discrete(θ_i)
    Draw $w_{ij} \sim$ Discrete($\phi_{z_{ij},\cdot}$)
  end for
end for

The key difference between Supervised-LDA and LDA is that for each training document, we first draw the label y and then choose a class-dependent Dirichlet prior for the topic proportions. The Dirichlet prior over the document specific topic proportions is represented as a $C \times K$ matrix $\boldsymbol{\alpha}$, where the c^{th} row of $\boldsymbol{\alpha}$ matrix corresponds to the Dirichlet prior for class c. Note that we consider both $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ (defined in Table 1) to be asymmetric Dirichlet priors. The number of words in each document, N_i, is an ancillary variable and we assume that it is independent of the class c. The Supervised-LDA model may also be viewed as a special case of models such as Labeled-LDA model [10] and Dirichlet-Multinomial Regression model [13].

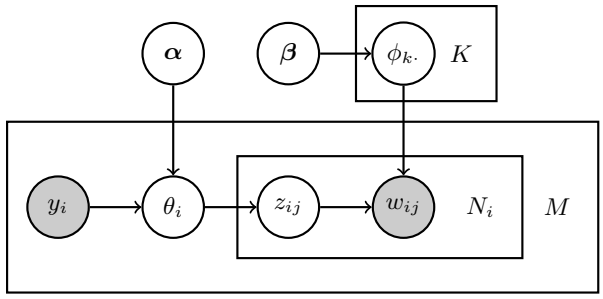

Fig. 1. Graphical model for Supervised-LDA

4. PARAMETER ESTIMATION IN SUPERVISED-LDA

Parameter estimation in Supervised-LDA is based on the maximum marginal likelihood principle. The marginal likelihood of the data, i.e., the likelihood of $(\mathbf{W}, \mathbf{Y})$ conditioned on the hyperparameters, is given by

$$p(\mathbf{W}, \mathbf{Y}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \int_{\boldsymbol{\theta}} \int_{\boldsymbol{\phi}} \sum_{\mathbf{Z}} p(\mathbf{W}, \mathbf{Y}, \mathbf{Z}, \boldsymbol{\theta}, \boldsymbol{\phi}|\boldsymbol{\alpha}, \boldsymbol{\beta})\, d\boldsymbol{\theta} d\boldsymbol{\phi},$$

where $\mathbf{Z}$ corresponds to the topic assignments of all the words in the training corpus. The above integral is intractable. Deterministic approaches (such as VB) replace the integral with a tractable lower bound. Sampling based approaches (such as CGS) approximate this integral (expectation) using an empirical (sample-based) average. The MAP estimation procedure approximates the integral by using point estimates of $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ ($\mathbf{Z}$ can be marginalized out)[1].

To the best of our knowledge, only VB inference has been explored earlier for the Supervised-LDA model [1]. We derive the update equations for MAP and CVB0 inference methods for the Supervised-LDA model. Not surprisingly, if we set all the $\mathbf{Y}$ to be equal (to 1) in our update equations, we recover the update equations for (unsupervised) LDA.

4.1. MAP estimation

The MAP estimate of $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ is given by

$$\boldsymbol{\theta}^*, \boldsymbol{\phi}^* = \arg\max_{\boldsymbol{\theta}, \boldsymbol{\phi}} p(\mathbf{W}, \boldsymbol{\theta}, \mathbf{Y}, \boldsymbol{\phi}|\boldsymbol{\alpha}, \boldsymbol{\beta}) \tag{1}$$

As shown in Appendix A, the objective function for MAP is given by

$$\begin{aligned}
&\log p(\mathbf{W}, \boldsymbol{\theta}, \mathbf{Y}, \boldsymbol{\phi}|\boldsymbol{\alpha}, \boldsymbol{\beta}) \\
&= \sum_{i=1}^{M} \Big(\sum_{v=1}^{V} \Big[n_{vi} \log(\boldsymbol{\phi}^\top \boldsymbol{\theta})_{vi} \Big] + \log P(y_i) \\
&\quad + \sum_{k=1}^{K} (\alpha_{y_i,k} - 1) \log \theta_{ik} - \log B(\alpha_{y_i,\cdot}) \Big) \\
&\quad + \sum_{k=1}^{K} \Big(\sum_{v=1}^{V} (\beta_v - 1) \log \phi_{kv} - \log B(\boldsymbol{\beta}) \Big),
\end{aligned} \tag{2}$$

[1] See Table 1 in [14] for a list of inference methods in graphical models.

where $B(\cdot)$ denotes the multinomial beta function. Parameter estimation is performed by maximizing (2) w.r.t. $\boldsymbol{\theta}$, $\boldsymbol{\phi}$, $\boldsymbol{\alpha}$, and $\boldsymbol{\beta}$ in a coordinate ascent fashion. The updates for θ_{ik} and ϕ_{kl} are given by,

$$\theta_{ik} \propto \max(g_{ik}, 0), \qquad \phi_{kv} \propto \max(h_{kv}, 0) \tag{3}$$

where $\sum_k \theta_{ik} = 1$, $\sum_v \phi_{kv} = 1$ and

$$g_{ik} = \hat{\theta}_{ik} \sum_{v=1}^{V} \Big[\phi_{kv} \frac{n_{vi}}{\sum_{k'=1}^{K} \phi_{k'v} \hat{\theta}_{ik'}} \Big] + (\alpha_{y_i,k} - 1),$$

$$h_{kv} = \hat{\phi}_{kv} \sum_{i=1}^{M} \Big[\theta_{ik} \frac{n_{vi}}{\sum_{k'=1}^{K} \theta_{ik'} \hat{\phi}_{k'v}} \Big] + (\beta_v - 1). \tag{4}$$

Note that $\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\phi}}$ denote the values of $\boldsymbol{\theta}, \boldsymbol{\phi}$ from the previous iteration.

4.1.1. Connection to NNMF

Note that (2) resembles the objective function of KL-divergence minimizing non-negative matrix factorization (NNMF) [15], with additional regularization terms on $\boldsymbol{\theta}, \boldsymbol{\phi}$. The equivalence between EM updates for Probabilistic Latent Semantic Analysis (PLSA) [16] and KL-divergence minimizing NNMF updates in the unregularized case (i.e., $\boldsymbol{\alpha} = 1, \boldsymbol{\beta} = 1$ in (2)), has been observed in [17]. In the EM algorithm for MAP solution in LDA [12], the E-step involves the computation of $\gamma_{wjk} = P(z_{ij}|w_{ij}, \theta_i)$ and the M-step involves maximization w.r.t. $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$. To ensure that γ_{wjk}'s are valid probabilities, [12] impose the constraint $\boldsymbol{\alpha} > 1, \boldsymbol{\beta} > 1$ in their MAP solution. Even if $\boldsymbol{\alpha} < 1, \boldsymbol{\beta} < 1$, γ_{wjk} can be valid probabilities if $g_{ik} \geq 0$ and $h_{kl} \geq 0$ in (4). Another subtle difference exists. In (2), the hyperparameters are optimized using Maximum likelihood (ML) estimation for Dirichlet distribution whereas in the MAP solution by [12], the hyperparameters are optimized using ML for Polya distribution. The Polya distribution accounts for the number of words in the document (and hence the number of topic variables), whereas the Dirichlet distribution estimates $\boldsymbol{\alpha}$ using the $\theta_i's$ and hence, does not account for document length in the hyperparameter estimation.

4.2. Collapsed Variational Bayes (CVB0)

In this section, we present the collapsed variational updates for Supervised-LDA. As in LDA, the collapsed variational distribution is assumed to factorize as follows[2] [12, 14]:

$$q(\mathbf{Z}, \boldsymbol{\theta}, \boldsymbol{\phi}) = q(\boldsymbol{\theta}, \boldsymbol{\phi}|\mathbf{Z}) \prod_i \prod_j q(z_{ij}), \tag{5}$$

where $q(z_{ij})$ is a multinomial distribution with parameters given by $q(z_{ij} = k) = \gamma_{ijk}$. The log marginal likelihood is lower bounded by the negative collapsed variational free energy [14, 18], i.e.,

$$\log p(\mathbf{W}, \mathbf{Y}|\boldsymbol{\alpha}, \boldsymbol{\beta}) \geq E_q[\log p(\mathbf{W}, \mathbf{Y}|\boldsymbol{\alpha}, \boldsymbol{\beta})] + H[q], \tag{6}$$

where the RHS denotes the negative collapsed variational free energy. Maximizing the zeroth order Taylor approximation to the negative collapsed variational free energy (hence the name CVB0), the updates for γ_{ijk} are obtained as

$$\gamma_{ijk} \propto (n_{ki}^{\setminus ij} + \alpha_{y_i,k}) \frac{n_{kv}^{\setminus ij} + \beta_v}{n_{k\cdot}^{\setminus ij} + \mathbf{1}^\top \boldsymbol{\beta}} \tag{7}$$

[2]To keep the notation uncluttered, we do not mention the variational parameters explicitly.

where $n_{kv}^{\setminus ij} = \sum_{(i',j') \notin (i,j), w_{i'j'} = v} \gamma_{i'j'k}$, $n_{ki}^{\setminus ij} = \sum_{j' \neq j} \gamma_{ij'k}$, and $n_{k\cdot}^{\setminus ij} = \sum_{(i',j') \notin (i,j)} \gamma_{i'j'k}$. Further details are available in Appendix B.

Note that in VB, a fully factorized variational distribution is assumed, i.e., $q(\mathbf{Z}, \boldsymbol{\theta}, \boldsymbol{\phi}) = \prod_i q(\theta_i) \prod_{ij} q(z_{ij}) \prod_k q(\phi_{k\cdot})$ and parameter estimation is performed by coordinate ascent on the negative variational free energy [1]. Upon convergence, the parameter estimates are computed using $\hat{\phi}_{kv} \propto (n_{kv} + \beta_v)$, where $\sum_v \hat{\phi}_{kv} = 1$. Note that while MAP is inherently parallelizable i.e., the θ_i's for the documents can be updated in parallel, the collapsed inference methods are not inherently parallelizable.

5. CLASSIFICATION

For a test document $\mathbf{w}_t$, model-based classification is made using the MAP rule, i.e.,

$$y_t^* = \arg\max_{y_t} \; p(y_t|\mathbf{w}_t) = \arg\max_{y_t} \; p(y_t, \mathbf{w}_t). \tag{8}$$

5.1. Classification using VB

VB can be used to classify a test document as follows [1]

$$y_t^* = \arg\max_{c} \; p(\mathbf{w}_t, y_t = c).$$

Since the RHS is intractable, they compare the variational lower bounds for $\log p(\mathbf{w}_t, y_t = c)$. The variational lower bound is computed as follows:

$$\log p(\mathbf{w}_t, y_t = c) \geq E_q[\log p(\mathbf{w}_t, y_t = c)] + H[q], \tag{9}$$

where q denotes the fully factorized variational distribution $q(\theta_t, \mathbf{z}_t) = q(\theta_t) \prod_{j=1}^{N_t} q(z_{tj})$. This approach requires recomputation of the variational lower bound for each possible value of y_t, and can be computationally demanding when C is large. Next, we present the classification rules for MAP and CVB0.

5.2. Classification using MAP

We express $p(y_t|\mathbf{w}_t)$ as follows

$$y_t^* = \arg\max_{y_t} \int_{\theta_t} p(\theta_t|\mathbf{w}_t) p(y_t|\theta_t) \; d\theta_t. \tag{10}$$

Since the integral in (10) cannot be computed in closed-form, we propose to approximate it as follows

$$y_t^* \approx \arg\max_{y_t} \; p(y_t|\theta_t^*) \int_{\theta_t} p(\theta_t|\mathbf{w}_t) \; d\theta_t \tag{11}$$

$$= \arg\max_{y_t} \; p(y_t|\theta_t^*), \tag{12}$$

where,

$$\theta_t^* = \arg\max_{\theta_t} \; p(\theta_t|\mathbf{w}_t), \tag{13}$$

$$= \arg\max_{\theta_t} \; \log p(\mathbf{w}_t|\theta_t) + \log p(\theta_t). \tag{14}$$

The approximation in (11) may be interpreted as a zeroth order version of Laplace approximation of the integral in (10) around θ_t^*. Additionally, obtaining a single θ_t^* (rather than C) enables us to think of Supervised-LDA as a supervised dimensionality reduction method.

Note that θ_t^* can be obtained by using an optimization similar to (2). Since y_t is unobserved for test data, we have $\log p(\theta_t)$ which is a mixture of Dirichlet distributions, instead of $\log p(\theta_i|y_i)$ used in training (2). We treat y_t as a latent variable and derive an EM algorithm to compute θ_t^*. The update rule is given by $\theta_{tk} \propto \max(g_{tk}, 0)$, where $\sum_k \theta_{tk} = 1$, and

$$g_{tk} = \sum_{l=1}^{V} \Big[n_{lt} \frac{\phi_{kl} \hat{\theta}_t(k)}{\sum_{u=1}^{K} \phi_{ul} \hat{\theta}_t(u)} \Big] + \sum_{c=1}^{C} P(y_t{=}c|\hat{\theta}_t)(\alpha_{ck} - 1). \quad (15)$$

Note that $\hat{\theta}_t$ denotes the value from the previous iteration. Note the similarity of (15) to (4). Since y_t is not observed, the $(\alpha_{ck} - 1)$ term is weighted by $P(y_t{=}c|\hat{\theta}_t)$.

5.3. Classification using CVB0

We consider two classification rules for CVB0. First, we classify y_t^* using

$$y_t^* = \arg\max_c \; p(\mathbf{w}_t, y_t = c).$$

Since the RHS is intractable, we use the collapsed variational lower bound for $\log p(\mathbf{w}_t, y_t = c)$. The collapsed variational lower bound is computed using (9), where the expectation is w.r.t the collapsed variational distribution, $q(\theta_t, \mathbf{z}_t) = q(\theta_t|\mathbf{z}_t) \prod_{j=1}^{N_t} q(z_{tj})$. We refer to this classifier as CVB0-1. Note that this can be computationally intensive when C is large. We introduce a second approach to alleviate this problem,

$$\gamma_{tjk} \;\propto\; (n_{kt}^{\setminus ij} + \sum_{c=1}^{C} P(y_t = c|\hat{n}_{\cdot t})\; \alpha_{c,k})\; \hat{\phi}_{kv}, \qquad (16)$$

where $n_{kt}^{\setminus tj} = \sum_{j' \neq j} \gamma_{tj'k}$, $\hat{n}_{\cdot t}$ denotes the value of $n_{\cdot t}$ from the previous iteration and $\hat{\phi}$ denotes the estimate of ϕ computed from the training data. We will refer to the second approximation as CVB0-2.

6. EXPERIMENTAL RESULTS

In the first experiment, we compare the classification accuracy and runtime achieved by MAP, CVB0, VB and PLSA for the Supervised-LDA model. We implemented CGS and observed that the runtime associated with CGS is significantly larger than the runtime associated with the other methods. In CGS, we need to compute the topic probabilities for each occurrence of a word (and not just every unique occurrence as in the other methods) as well as draw multiple topic samples before estimating the hyperparameters. Hence, we do not include the results obtained using CGS here.

6.1. Implementation details

All the methods were implemented in Matlab. We used similar vectorization techniques in all of our implementations. We plan to make our code publicly available in the near future.

6.1.1. Hyperparameter optimization for MAP

We restricted $\boldsymbol{\alpha} \geq 1$, but did not impose any constraint on $\boldsymbol{\beta}$. In our experiments, we used a log-barrier method with Newton update equations [19] to compute the optimum $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ in (2).

6.1.2. Hyperparameter optimization for CVB0

We used the fixed point updates in [20] to compute the ML estimates of Polya distribution.

6.1.3. Hyperparameter optimization for VB

As done by [1], we optimize $\boldsymbol{\alpha}$, but set $\boldsymbol{\beta}$ to 1.

6.1.4. PLSA-NN

As observed by [21], PLSA is equivalent to LDA when all the hyperparameters are set to 1. Hence, we used the same update equations as that of the MAP estimation, except that the hyperparameters are held constant at 1. Once the θ's have been obtained for the training as well as the test data, we use a k-nearest neighbor (k-NN) classifier with Euclidean distance metric for classifying the test documents. Following [22], we set the number of nearest neighbors, $k = 10$.

6.2. Datasets

In this section, we describe the datasets used in our experiments.
LabelMe: The first image classification dataset was used in [3]. The dataset consists of 1600 images from the *LabelMe* toolbox. There are totally eight classes. For each image, the Scale-invariant feature transform (SIFT) vectors are computed and then the SIFT vectors are clustered to obtain the codebook (size=158) representation. Each image contains 2401 SIFT vectors. The pre-processed dataset in bag-of-words format was made publicly available by [3]. More details regarding the dataset are available in [3]. We used three random train-test data splits for cross validation, each time dividing the data into 800 training documents and 800 test documents.
MSRC-v2: We evaluate our algorithms on a subset of the MSRC-v2 dataset[3]. We used images belonging to eight groups (i.e., class labels), namely, `'book'`, `'grass, cow'`, `'tree, grass, sky'`, `'bike, building'`, `'sign'`, `'water, boat'`, `'aeroplane, grass, sky'`, `'road, building'` resulting in a total of 240 images. We divided each image into 8×8 blocks and we cluster the blocks using k-means algorithm to create a codebook of size 160. Using this codebook, we create the bag-of-words representation for each image. Again, we use 50% of the dataset for training and 50% of the dataset for testing. While we realize that it might be possible to use more sophisticated features, our goal here is to compare inference methods for classification rather than find good feature vector representations. Note that quite some overlap exists between the classes themselves.

6.3. Simulation details

We vary the number of topics and report the classification accuracy and runtime in each case. Note that, in practice, we are interested only in the performance at the optimal K (chosen based on a validation set) for each method. For each train-test data split, we try three random initializations and report the best classification accuracy and the total runtime. The total runtime is the sum of the runtimes for each random initialization, which is the sum of the time for training and the time for testing. We compute the mean and standard deviation of the results based on the 3 random train-test splits. The error bars in our graphs denote the variation amongst 3 random train-test splits for cross-validation. We train the model till the fractional change in log likelihood, given by $abs[(ll_{new} - ll_{old})/ll_{new}]$, is less

[3] http://research.microsoft.com/en-us/projects/objectclassrecognition/

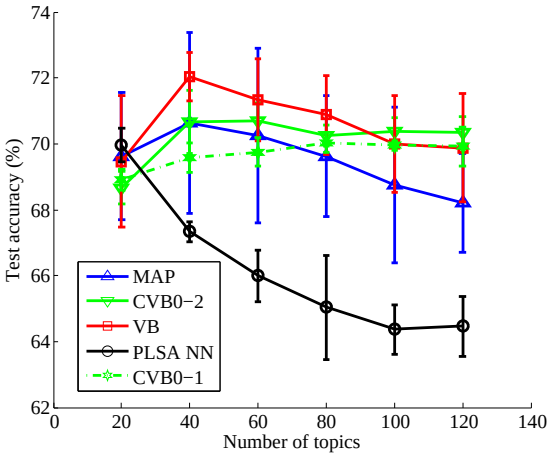

Fig. 2. *LabelMe dataset:* Comparison of classification accuracy obtained using MAP, CVB0-1, CVB0-2, VB and PLSA-NN.

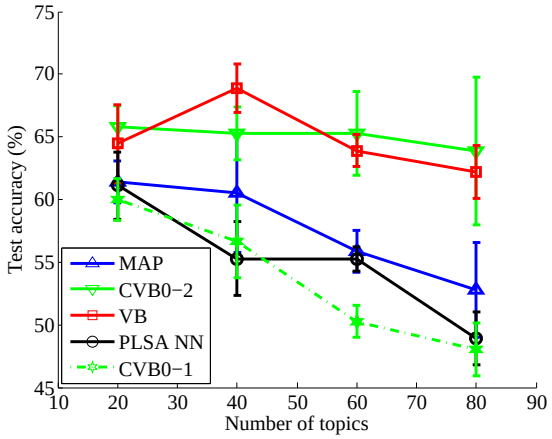

Fig. 3. *MSRC-v2 dataset:* Comparison of classification accuracy obtained using MAP, CVB0-1, CVB0-2, VB and PLSA-NN.

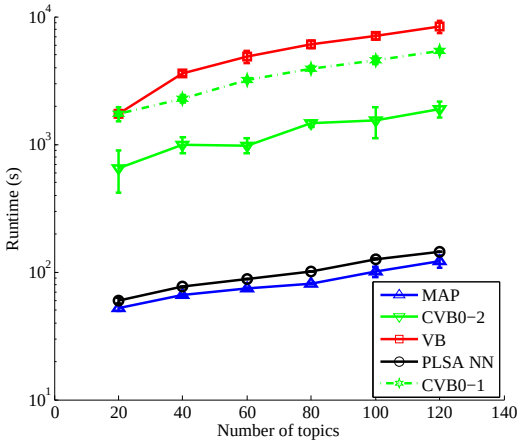

Fig. 4. *LabelMe dataset:* Comparison of run time of MAP, CVB0-1, CVB0-2, VB and PLSA-NN.

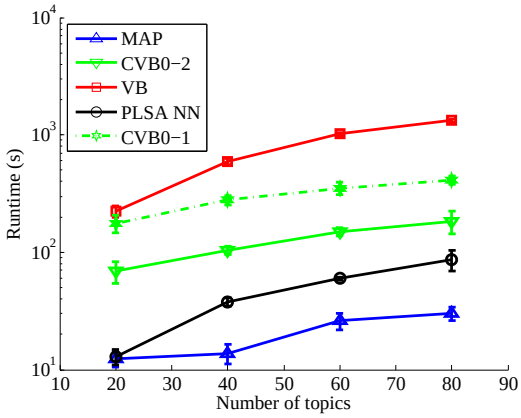

Fig. 5. *MSRC-v2 dataset:* Comparison of run time of MAP, CVB0-1, CVB0-2, VB and PLSA-NN.

than a threshold (10^{-6} in our experiments), with an additional limit on the maximum number of iterations (300 in our experiments).

6.4. Effect of the inference method on classification accuracy

The classification accuracy comparison is shown in Fig. 2 and Fig. 3 for the LabelMe and MSRC-v2 datasets respectively. We can observe that MAP provides comparable performance to VB, CVB0 in terms of classification accuracy in the LabelMe dataset. However, in the MSRC-v2 dataset, CVB0 outperforms MAP and performs quite similar to VB. The CVB0-2 classifier provides comparable performance to CVB0-1 classifier.

6.5. Computational complexity

The runtime comparison for the LabelMe and MSRC-v2 datasets are shown in Fig. 4 and Fig. 5 respectively. We observe that MAP provides considerable advantage in terms of runtime. The CVB0-2 classifier is significantly faster than CVB0-1. We observe the trend MAP $<$ CVB0-2 $<$ CVB0-1 $<$ VB. To gain further insight, we consider the computational complexity of the updates and the implementation aspects. During the classification stage, CVB0-2 and MAP are roughly $O(C)$ times faster than VB and CVB0-1. The computational complexity in the training stage for Supervised-LDA is similar to that of LDA (see Section 5 in [12] for a related discussion). Let $U = \sum_i U_i$, where $U_i = |\{v \in \{1, 2, \ldots, V\}, n_{vi} \neq 0\}|$, i.e., U_i denotes the number of unique words in the i^{th} document. Note that usually $U < VM$. Let $N = \sum_i N_i$, where N_i denotes the number of words in the i^{th} document, as defined in Table 1. The training complexity per iteration is $O(KU)$ for MAP, CVB0 and VB and $O(KN)$ for CGS. As mentioned earlier, MAP updates for different documents can be parallelized while the collapsed inference methods such as CVB and CGS are not inherently parallelizable. VB is slower than CVB0-1 due to expensive digamma computations. In our experiments (not reported here), we have observed that CGS is slower than VB due to the large number of iterations required.

The computational complexity of PLSA and MAP seems to be comparable in these datasets. However, for classification using PLSA, we need to compute the nearest neighbor for the test data, which can be computationally intensive for large scale applications.

7. CONCLUSION

We presented MAP and CVB0 inference methods for the Supervised-LDA model. We introduced a computationally efficient classification algorithm for MAP and CVB0 that is scalable for datasets involving large number of classes. Additionally, this classification algorithm allows us to use Supervised-LDA as a supervised dimensionality reduction tool. We provided an empirical comparison of the classification accuracy and runtime of MAP, CVB0 to VB. The results indicate that, with proper hyperparameter tuning, CVB0 and VB can yield similar classification performance, while MAP yields a slightly lower performance. However, MAP is computationally very efficient and can provide speed-ups of over an order of magnitude compared to VB and CVB0. Based on our results, we advocate CVB0 parameter estimation with the CVB0-2 classifier for the Supervised-LDA model, since it provides a good tradeoff between classification accuracy and run time. Future work will explore the extension of our inference methods to more complex topic models that can handle annotations and multiple labels.

8. REFERENCES

[1] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *Proc. CVPR 2005*, 2005, pp. 524–531.

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

[3] C. Wang, D. Blei, and L. Fei-Fei, "Simultaneous image classification and annotation," in *Proc. CVPR*, 2009.

[4] L. Du, L. Ren, B. Dunson, and L. Carin, "A Bayesian Model for Simultaneous Image Clustering, Annotation and Object Segmentation," *Advances in NIPS*, 2009.

[5] S. Lacoste-Julien, F. Sha, and M.I. Jordan, "DiscLDA: Discriminative learning for dimensionality reduction and classification," *Advances in NIPS 21*, 2008.

[6] J. Zhu, A. Ahmed, and E.P. Xing, "MedLDA: maximum margin supervised topic models for regression and classification," in *Proc. ICML*. ACM New York, NY, USA, 2009.

[7] S.H. Yang, H. Zha, and B.G. Hu, "Dirichlet-Bernoulli Alignment: A Generative Model for Multi-Class Multi-Label Multi-Instance Corpora," *Advances in NIPS*, 2009.

[8] D.M. Blei and J. McAuliffe, "Supervised topic models," *Advances in NIPS*, vol. 20, pp. 121–128, 2008.

[9] G. Xu, S.H. Yang, and H. Li, "Named entity mining from click-through data using weakly supervised latent dirichlet allocation," in *Proc. ACM SIGKDD*. ACM New York, NY, USA, 2009, pp. 1365–1374.

[10] 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. EMNLP*, 2009, pp. 248–256.

[11] H.M. Wallach, D. Mimno, and A. McCallum, "Rethinking LDA: Why priors matter," in *Topic Models: Text and Beyond Workshop in NIPS Conference*, 2009.

[12] A. Asuncion, M. Welling, P. Smyth, and Y.W. Teh, "On smoothing and inference for topic models," in *UAI*, 2009.

[13] D. Mimno and A. McCallum, "Topic models conditioned on arbitrary features with dirichlet-multinomial regression," in *Proc. UAI*, 2008.

[14] Y.W. Teh, D. Newman, and M. Welling, "A collapsed variational bayesian inference algorithm for latent dirichlet allocation," *Advances in NIPS*, vol. 19, pp. 1353, 2007.

[15] D.D. Lee and H.S. Seung, "Algorithms for non-negative matrix factorization," *Advances in NIPS*, vol. 13, 2001.

[16] T. Hofmann, "Probabilistic latent semantic indexing," in *Proc. ACM SIGIR*. ACM, NY, USA, 1999, pp. 50–57.

[17] E. Gaussier and C. Goutte, "Relation between PLSA and NMF and implications," in *Proc. ACM SIGIR*. ACM, 2005, pp. 601–602.

[18] M.J. Beal, *Variational algorithms for approximate Bayesian inference*, 2003.

[19] S.P. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge University Press, 2004.

[20] T. Minka, "Estimating a Dirichlet distribution," 2003.

[21] M. Girolami and A. Kabán, "On an equivalence between PLSI and LDA," in *Proc. ACM SIGIR*. ACM, 2003, p. 434.

[22] A. Bosch, A. Zisserman, and X. Munoz, "Scene classification via pLSA," *ECCV 2006*, pp. 517–530, 2006.

A. DERIVATION OF MAP OBJECTIVE FUNCTION

Based on the structure of the graphical model, we have

$$p(\mathbf{W}, \boldsymbol{\theta}, \mathbf{Y}, \boldsymbol{\phi}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = p(\boldsymbol{\phi}|\boldsymbol{\beta}) \prod_{i=1}^{M} p(\mathbf{w}_i, \theta_i, y_i|\boldsymbol{\alpha}, \boldsymbol{\phi}). \quad (17)$$

Next, we derive the expressions for each term on the RHS of (17). Let $\mathbf{z}_i$ denote the respective topic assignments for each word in the vector $\mathbf{w}_i$. For each document, we can marginalize out all possible topic assignments, i.e.,

$$p(\mathbf{w}_i, \theta_i, y_i|\boldsymbol{\alpha}, \boldsymbol{\phi}) = \Big(\sum_{\mathbf{z}_i} P(\mathbf{w}_i|\mathbf{z}_i, \boldsymbol{\phi}) P(\mathbf{z}_i|\theta_i)\Big) p(\theta_i|y_i, \boldsymbol{\alpha}) P(y_i).$$

The first term in the RHS of the above equation can be simplified as

$$\sum_{\mathbf{z}_i} P(\mathbf{w}_i|\mathbf{z}_i, \boldsymbol{\phi}) P(\mathbf{z}_i|\theta_i) = \exp(\sum_{v=1}^{V} n_{vi} \log(\boldsymbol{\phi}^\top \boldsymbol{\theta})_{vi}), \quad (18)$$

where n_{vi} denotes the number of times word v occurs in the i^{th} training document. Using the fact that $p(\theta_i|y_i, \boldsymbol{\alpha})$ and $p(\phi_{k,\cdot}|\boldsymbol{\beta})$ are Dirichlet distributions, we obtain (2).

B. DERIVATION OF CVB0 UPDATES

The RHS of (6) can be expanded as follows

$$E_{q(\mathbf{Z})}[E_{q(\boldsymbol{\theta},\boldsymbol{\phi}|\mathbf{Z})}[\log p(\mathbf{W}, \mathbf{Y}, \mathbf{Z}, \boldsymbol{\theta}, \boldsymbol{\phi}|\boldsymbol{\alpha}, \boldsymbol{\beta})] + H[q(\boldsymbol{\theta}, \boldsymbol{\phi}|\mathbf{Z})]] + H[q(\mathbf{Z})].$$

Maximizing the above expression w.r.t γ_{ijk}, we obtain

$$\begin{aligned} \gamma_{ijk} \quad \propto \quad & \exp(E_{q(\mathbf{Z}^{\setminus ij})}[\log(n_{ki}^{\setminus ij} + \alpha_{y_i,k}) + \log(n_{kv}^{\setminus ij} + \beta_v) \\ & - \log(n_{k\cdot}^{\setminus ij} + \mathbf{1}^\top \boldsymbol{\beta})]) \end{aligned}$$

where $\sum_k \gamma_{ijk} = 1$. Using the zeroth order Taylor approximation in the above expectations [12], we obtain (7).