Improving Gibbs Sampling Predictions on Unseen Data for Latent Dirichlet Allocation

Yannis Papanikolaou ypapanik@csd.auth.gr
Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece

Timothy N. Rubin timrubin@indiana.edu
Cognitive Computing Laboratory
Indiana University
Bloomington, IN, USA

Grigorios Tsoumakas greg@csd.auth.gr
Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece

Editor: tba

Abstract

Latent Dirichlet Allocation (LDA) is a model for discovering the underlying structure of a given data set. LDA and its extensions have been used in unsupervised and supervised learning tasks across a variety of data types including textual, image and biological data. Several methods have been presented for approximate inference of LDA parameters, including Variational Bayes (VB), Collapsed Gibbs Sampling (CGS) and Collapsed Variational Bayes (CVB) techniques. This work explores three novel methods for generating LDA predictions on unobserved data, given a model trained by CGS. We present extensive experiments on real-world data sets for both standard unsupervised LDA and Prior LDA, one of the supervised variants of LDA for multi-label data. In both supervised and unsupervised settings, we perform extensive empirical comparison of our prediction methods with the standard predictions generated by CGS and CVB0 (a variant of CVB). The results show a consistent advantage of one of our methods over CGS under all experimental conditions, and over CVB0 under the majority of conditions.

Keywords: Latent Dirichlet Allocation, unsupervised learning, multi-label classification, text mining, Collapsed Gibbs Sampling, CVB0

1. Introduction

The Latent Dirichlet Allocation (LDA) model was introduced by Blei et al. (2003) over a decade ago. Since then, LDA has been applied to numerous tasks, such as text mining (AlSumait et al., 2008), computer vision (Cao and Fei-Fei, 2007), social-network analysis (Zhang et al., 2007) and bio-informatics (Zheng et al., 2006). It has also been the subject of numerous adaptations and improvements to deal with, for example, supervised learning tasks (Blei and McAuliffe, 2007; Ramage et al., 2009; Zhu et al., 2009), time and memory efficiency issues (Porteous et al., 2008; Yao et al., 2009; Newman et al., 2007) and large or streaming data settings (Hoffman et al., 2010b; Gohr et al., 2009; Rubin et al., 2012).

The present paper focuses on the situation in which an LDA model has already been trained, and we wish to make predictions about unobserved data. This situation arises in both unsupervised and supervised learning settings. In the unsupervised setting, a model is trained on some initial data, wherein the relevant parameters of interest (e. g. the topic parameters) are estimated. Then, based on this estimated model, we make predictions about future data, such as which of the already determined topics better describe a new document, or what additional words we would expect to see in a partially observed document. Similarly, in the multi-label learning setting — more specifically in the Prior LDA model (Rubin et al., 2012) that we consider here — during testing, the document-wise parameters are inferred for all test documents. These document-wise parameters are then used to assign labels to these documents. In both of cases, estimation of the document-level model parameters is the key to generating the predictions of interest.

The contributions of this work can be summarized as follows:

- We propose three alternative methods for estimating the document-level parameters of an LDA model that has been trained using collapsed Gibbs sampling (CGS)(Griffiths and Steyvers, 2004).
- We present an extensive empirical comparison of the predictions generated by our three proposed methods against the predictions of standard CGS as well as CVB0 (Asuncion et al., 2009) — a variant of the collapsed variational Bayes (CVB) method — in both unsupervised and supervised learning settings.
- We provide additional experimental comparisons of the CGS and CVB0 algorithms regarding their convergence behavior, to further contextualize our empirical results.

As this paper deals with LDA in both an unsupervised and a supervised learning setting, for concision we will treat the following terms as equivalent when describing the procedure of fitting an LDA model from training data: 'estimation', 'fitting' and 'training'. Similarly, the terms 'prediction' and 'inference' will be applied to the case of predicting on test data (that was unobserved during training). Additionally, since Prior LDA is essentially a special case of unsupervised LDA, we will focus our model descriptions on the case of standard LDA, and specify the exceptions as they apply to Prior LDA. We furthermore note here that we will be using the Collapsed Gibbs Sampling (CGS) algorithm described by Griffiths and Steyvers (2004)—which approximates the LDA parameters of interest through iterative sampling in a MCMC procedure—unless otherwise specified.

The rest of the paper is organized as follows: Section 2 describes the LDA and Prior LDA models, and gives an overview of the related literature. Section 3 introduces the alternative prediction methods for LDA that we propose. Section 4 describes our experiments and the corresponding results in unsupervised learning tasks, comparing our prediction methods to the CGS and CVB0 predictions. In Section 5, the second set of experiments is described; we conduct the same comparison of our methods to CGS and CVB0 when these algorithms are applied to the Prior LDA model in multi-label classification scenarios. Finally, we draw the relevant conclusions and refer to some possible extensions of this work in Section 6.

2. Background and Related Work

LDA is a hierarchical Bayesian model that describes how a corpus of data is generated via a set of unobserved *topics*. For convenience in describing this model, we will assume that we are dealing with a corpus of unlabeled documents. The LDA model assumes that there exists some unobserved set of topics, L, where each topic is parameterized by a multinomial probability distribution over words, which captures some coherent semantic theme within the corpus. Furthermore, LDA assumes that each document can be described as a multinomial distribution over these topics. The process for generating each document then involves first sampling a topic from the document's distribution over topics, and then sampling a word from the corresponding topic's distribution over words. More formally, the process for generating a corpus of documents is described by LDA as follows:

- For each topic l, sample a multinomial distribution ϕ_l over the words of the corpus from a Dirichlet(β)
- For each document d, sample a multinomial distribution θ_d over topics from a Dirichlet(α)
- For each of the word tokens w_i in d:
  - Sample a topic z_i from θ_d
  - Sample a word type w from ϕ_{z_i}.

The goal of inference is to estimate the θ and ϕ parameters—the multinomial distributions for documents over topics, and topics over words, respectively. In doing so, we learn a lower-dimensional representation of the structure of all documents in the corpus in terms of the topics.

In the unsupervised learning context, this lower-dimensional representation of documents is useful for both summarizing documents and for generating predictions about future, unobserved documents. In the supervised, multi-label learning context, extensions of the basic LDA model—such as Prior LDA — put topics into one-to-one correspondence with labels, and the model is used for assigning labels to test documents.

Formally, let us denote as V the vocabulary (the set of unique word-types) and as D the set of documents in the corpus. We will use L to denote the set of topics. Similarly, w denotes a word token, d a document, and l a topic (or in the case of Prior LDA, a label). The parameter $\phi_l(w)$, $l \in L$, $w \in V$ represents the multinomial distribution of topic (or label) l over words, and $\theta_d(l)$ represents the multinomial distribution over topics (or labels, in Prior LDA) for document d. Moreover, α will denote the Dirichlet prior on θ and β the Dirichlet prior on ϕ. The term V_d denotes the document-specific vocabulary (the word types) and N_d the number of word tokens of a given document d. Lastly, in the multi-label scenario, L_d stands for the set of labels that were assigned to d.

During sampling, Collapsed Gibbs Sampling (CGS) updates the hard-assignment z_i of a word w_i to one of the topics $l \in L$. This update is performed sequentially for all word tokens in the corpus, and then this process is repeated until the algorithm has converged.

The CGS algorithm makes use of count matrices during sampling. We will employ the following notation for these count matrices: $n_{w_i,l}$ represents the number of times that w_i

V	the set of features, consisting of unique word-types
L	the set of topics (or labels)
D	the set of documents
w, w_i	a single word token, where $w_i \in V$
d	a document
l	a topic (label)
z_i	the topic assignment to a word w_i
$\phi_l(w)$	topic-word distributions
$\theta_d(l)$	document-topic distributions
V_d	features of d
N_d	word tokens of d
L_d	label of d
$n_{w_i,l}$	# of times that l is assigned to w_i across the corpus
$n_{d,l}$	# of features in d that have l assigned to them

Table 1: Notation used throughout the article

is assigned to topic l across the corpus, and $n_{d,l}$ represents the number of word tokens in document d that have been assigned to topic l. Table 1 summarizes the notation used along the paper.

The update equation giving the probability of assigning z_i to topic l, conditional on w_i, d, the hyper-parameters α and β, and the current topic assignments of all other words (represented by $\cdot$) is:

$$p(z_{w_i} = l |\, w_i,\, d,\, \boldsymbol{z_{\neg i}},\, \alpha, \beta, \cdot) = \frac{n_{w_i,l\neg i} + \beta}{\sum\limits_{w \in V} (n_{w_i,l\neg i} + \beta)} \times \frac{n_{d,l\neg i} + \alpha}{\sum\limits_{d \in D} (n_{d,l\neg i} + \alpha)} \tag{1}$$

In the above equation, one subtracts from all count matrices n the current assignment of w_i, as indicated by the $\neg i$ notation. After the algorithm has converged, a final point estimate of the probability of word w given topic l is computed as:

$$\phi_l(w) = \frac{n_{w,l} + \beta}{\sum\limits_{w \in V} (n_{w,l} + \beta)} \tag{2}$$

Similarly, a point estimate for the probability of the topic l given document d is given by

$$\theta_d(l) = \frac{n_{d,l} + \alpha}{\sum\limits_{d \in D} (n_{d,l} + \alpha)} \tag{3}$$

We provide the pseudocode for training an LDA model using the CGS algorithm (Alg. 1), in order to clarify how and where the methods we propose are employed. For a more detailed discussion of CGS, see (Griffiths and Steyvers, 2004).

During prediction (when performing CGS on documents that were unobserved during training), the only difference is that the ϕ distributions are fixed and equal to the ones learned during estimation; The sampling update presented in Equation 1 thus becomes:

$$p(z_{w_i} = l|\, w_i,\, d,\, \boldsymbol{z_{\neg i}},\, \cdot) = \phi_l(w) \times \frac{n_{d,l\neg i} + \alpha}{\sum\limits_{d \in D} (n_{d,l\neg i} + \alpha)} \tag{4}$$

The only posterior distributions to be computed in this case are the θ_d.

Algorithm 1 Gibbs Sampling for the LDA

```
for all documents $d \in D$ do
    for each word $w_i \in d$ do
        assign randomly a topic $l \in L$ to $w_i$ such that $z(w_i, d) \leftarrow l$
        update $n_{w,l}$, $n_{d,l}$ accordingly
    end for
end for
for each iteration $i$ do
    for each $d \in D$ do
        for each $w_i \in d$ do
            for each $l \in L$ do
                calculate $p(w_i, d, l)$ according to Equation 1
            end for
            sample a topic assignment $z(w_i, d) \sim Conditional(p(w_i, d))$
            update $n_{w,l}$, $n_{d,l}$ accordingly
        end for
    end for
    if $i \, mod \, samplingInterval = 0$ then
        calculate a point estimate of $\phi_l$ according to Equation 2
        calculate a point estimate of $\theta_d$ according to Equation 3
    end if
end for
```

An extension to unsupervised LDA for supervised multi-label document classification was proposed by Ramage et al. (2009). Their algorithm, Labeled LDA (LLDA), applies a one-to-one correspondence between topics and labels. During estimation of the model, the possible assignments for a word token to a topic are constrained to be the topics corresponding to the training document's observed labels. Therefore, during training, the sampling update is equivalent to Equation 1, except that we are now assigning word tokens to 'labels', and the probability of assigning a word in document d to a label l is zero if the label was not assigned to d. Inference on test documents is performed similarly to standard LDA; estimates of the label-word distributions (ϕ_l) are learned on the training data and are fixed, and then the test documents' θ distributions are estimated. However, unlike in unsupervised LDA, where topics can change from iteration to iteration, in LLDA topics remain steady ('anchored' to a label) and therefore it is possible to average point estimates of ϕ and θ over multiple Markov chains, thereby improving performance.

Rubin et al. (2012) presented two extensions of the LLDA model. The first extension, Prior LDA, takes into account the label frequencies in the corpus via an informative Dirichlet prior over parameter θ. The second extension takes into account label dependencies by training a second-level LDA model on the observed label-tokens (Dependency LDA). Li et al. (2015) have presented two variations of the previous work by using the training corpus to directly observe label frequencies and dependencies likewise. Finally, Ramage et al. (2011) have introduced PLDA to relax the constraints of LLDA and exploit the unsupervised and supervised forms of LDA simultaneously; their algorithm attempts to model hidden topics within each label, as well as unlabeled, corpus-wide latent topics.

For one of the evaluation methods we consider, we employ a technique presented in (Yao et al., 2009). The authors proposed three prediction methods based on the Gibbs sampling algorithm (Griffiths and Steyvers, 2004) with the differences among them residing in how the already existing instances are combined with the unseen ones. The first method, Gibbs1, fits a model to the training data, and after convergence saves this model. Then, new instances are added, and the sampling procedure is repeated again until convergence. This approach forces an approximate alignment of the two sets of topics. The second step of this procedure is similar to fitting a new model for the complete data set, except that the approximate alignment of topics between the intermediate and final models allows one to directly compare the document-level parameters of these two models. We utilize this approach for evaluation of our unsupervised models, in addition to the more traditional "perplexity" measures.

The methods proposed in this paper are closely connected to an extension of the CVB (Teh et al., 2006) algorithm—CVB with zeroth-order information (CVB0) proposed by Asuncion et al. (2009). CVB0, along with its generalized form CVB, is a deterministic algorithm that combines the theoretical properties of both Variational Bayes and CGS. From an implementation perspective, CVB0 resembles the CGS algorithm, with the difference being that in every pass, instead of probabilistically sampling a hard topic-assignment for every token based on the sampling distribution in Equation (1), the algorithm keeps (and updates) that probability mass for every word token. A consequence of this procedure is that CVB0 is a deterministic algorithm, whereas CGS is a stochastic algorithm (Asuncion, 2010). In our approaches, we employ Gibbs sampling for our inference method, but then make final predictions from the model trained by CGS that more closely resemble the approach taken by the CVB0 update procedure. We find that this approach outperforms both CGS and CVB0 approaches for predictions of document topics (in unsupervised learning contexts) and document labels (in supervised learning contexts). A stochastic extension of CVB0, SCVB0, was presented by Foulds et al. (2013). Both SCVB0 and the Stochastic Variational Bayes algorithm presented by Hoffman et al. (2010a) focus on efficient and fast inference of the LDA parameters, to account for massive-scale data scenarios.

3. Inference Methods on Unseen Data

Having described the standard CGS methods for training and prediction, we proceed to the description of the three methods we consider for making predictions of the θ_d parameters on test data, given an LDA model trained using CGS. As explained before, the goal of the inference process when predicting on an unseen set of documents, D, is to estimate $|D|$ posterior multinomial distributions of documents over topics, namely the θ distributions. These distributions essentially supply a ranking of topics for each document, from the most relevant (the one with the highest probability $\theta_d(l)$) to the least relevant (the one with the lowest probability $\theta_d(l)$ respectively). Predictions from the CGS algorithm θ are typically calculated using Equation (3). However, there are alternative approaches for calculating θ that one could employ, such as using the estimator from the CVB0 algorithm.

We present three alternative methods for calculating θ at the time of prediction on unseen data: (a) summing over the z topic assignments for every document, (b) summing over the conditional probabilities p (Equation 4) for every document and (c) combining (a) and (b). In the following, we present the details of these approaches.

3.1 Averaging the z topic assignments

A first alternative strategy to Equation (3) for estimating θ_d, is to average over the z assignments for every document. This is equivalent to the standard estimate for θ_d, except the α smoothing priors have been removed:

$$\theta_z(d,l) = \frac{n_{d,l}}{\sum\limits_{d=1}^{D} n_{d,l}} \tag{5}$$

The motivation for this approach is that, although the smoothing priors are important during sampling, it is possible that at test time they simply smooth over some of the signal in the estimate of θ. If this is true, removing the smoothing prior α from the estimate of θ should improve the quality of predictions. We will refer to this method as CGS$_z$.

In case of Prior LDA, we initially considered an additional approach related to the one above. For this method, we took the mode topic-assignment, across multiple samples from the Markov Chain, for each individual token. So, e.g., if we used five samples, and the first token in a document was assigned to topic 1 twice, and topics 2 through 4 once, we would treat this as a single assignment of the token to topic 1. Subsequently, the relevant topics (or labels) for each document were determined as the set of mode topic-assignments of the document's word tokens. For instance, in a document with three word tokens, assigned to labels 2, 1 and 2 respectively, the labels-set characterizing the document would be {1, 2}. The motivation for this approach was to obtain directly a hard-assignment of labels for a document (rather than a distribution) and bypass the problem of determining a proper threshold to apply on the θ_d document-topics distribution. However, this method did not perform well in preliminary experiments, and is therefore not further considered within the paper.

3.2 Averaging the conditional probabilities

Another approach we consider is averaging over the conditional probabilities from Equation 1 for every document. As stated before, p expresses the probability of assigning to a word token a given topic, conditioned on the word, the document, and the other assignments of words to topics in both the given document (through n_{dl}) and in the whole corpus (through n_{wl} during training and ϕ during prediction). The idea is to average these word-level probabilities for every document and obtain a total probability $p(l|d)$ such that:

$$\theta_p(d,l) = p(l|d,\,\cdot) = \sum_{w=1}^{N_d} p(z_{w_i} = l|\, w_i,\, d) = \sum_{w=1}^{N_d} \phi_l(w) \times \frac{n_{d,l} + \alpha}{\sum\limits_{d=1}^{D} (n_{d,l} + \alpha)} \tag{6}$$

The motivation for this approach is that the conditional probabilities should provide a richer representation of the document-topics distribution. We note here, that as the probability $p(z_{w_i} = l|\, w_i,\, d)$ is unnormalized, an additional step of normalization is required so that $\sum\limits_{l=1}^{L} p(z_{w_i} = l|\, w_i,\, d) = 1$. We will refer to this approach as CGS$_p$.

During Section 2, we described briefly the CVB0 algorithm as being related to this method. In order to illustrate this resemblance we first consider more precisely the differences between CGS and CVB0: CGS uses a stochastic, iterative procedure in which (a) for each word of each document, a topic assignment z is sampled from the conditional probability, (b) the counts $n_{w,l}$ and $n_{d,l}$ are updated based on these assignments, and (c) the conditional probability is computed from Equation 1. CVB0 on the other hand, is not using the hard-assignments z of words to specific topics, but introduces another set of variables (γ in the original paper), which store the complete, normalized probability mass over the topics for each word token. Hence, in CVB0, CGS step 'a' is omitted and step 'b' is modified so that $n_{w,l}$ and $n_{d,l}$ store the respective sums of probabilities. From the above, it becomes clear that during inference on unseen data, when calculating θ, CVB0 employs a similar equation like CGS$_p$ in the sense that they both use the conditional probability instead of the word-topic assignments in order to calculate the topics-documents distributions. However, unlike CVB0, CGS$_p$ still employs CGS throughout inference (i.e., at all points except when we are computing our final estimate of θ).

3.3 Combining the z-assignments with the conditional probabilities

We also consider a variant of the previous method, CGS_p, in which we remove the α smoothing priors from Equation 6, as we did for the standard estimator of θ in Section 3.1.

$$\theta_{p|z}(d,l) = \sum_{w=1}^{N_d} \phi(w,l) \times \frac{n_{d,l}}{\sum\limits_{d=1}^{D} n_{d,l}} \tag{7}$$

We will refer to this approach as CGS$_{p|z}$. Table 2 lists the above methods, together with their main equation and their computational complexity for taking a sample.

Method	Equation	Complexity
CGS	3	$\mathcal{O}(D \times L)$
CGS$_z$	5	$\mathcal{O}(D \times L)$
CGS$_p$	6	$\mathcal{O}(D \times N_d \times L)$
CGS$_{p\|z}$	7	$\mathcal{O}(D \times N_d \times L)$

Table 2: Prediction methods for inference of the documents-topics distributions

4. Unsupervised learning experiments - LDA

This section describes the experiments that we performed for unsupervised LDA models. We first describe the data sets, the evaluation procedures and the relevant experimental setup. Subsequently, we present the relevant results followed by a brief discussion of the implications of our findings.

4.1 Data sets

Four data sets were used in the unsupervised setting: a) The BioASQ subset data set, b) the BioASQ labels data set, c) the Reuters corpus and d) the TASA subset corpus. The relevant statistics are shown in Table 3.

Data Set	Training Set	Test Set	Average Length	Word Types
BioASQ	19,000	1,000	113.55	22,394
BioASQ labels	19,000	1,000	13.85	16,311
Reuters	10,000	1,000	49.68	7,146
TASA	4750	250	107.62	8787

Table 3: Statistics for the data sets used in the unsupervised learning experiments. Average length and word types are given for the respective training sets.

4.1.1 BioASQ data sets

The BioASQ challenge (Balikas et al., 2014) deals with large-scale online multi-label classification of biomedical journal articles from MEDLINE , the premier bibliographic database of the National Library of Medicine in the United States. This supervised learning task is particularly challenging as the taxonomy of labels includes around 27000 terms, which have extremely imbalanced frequencies (Papanikolaou et al., 2015). Furthermore, for every article, only the title and the abstract are provided, not the full text.

For the purpose of the unsupervised experiments of this section, we used a small subset of the BioASQ corpus, consisting of $19,000$ training documents and $1,000$ test documents. We constructed two data sets out of this subset. In the first one, called *BioASQ*, we concatenated the abstract and title of each article and removed common stop-words, infrequent (less than 5 appearances in the corpus), and uninformative words (appearing in more than one quarter of the corpus). The remaining uni-grams were used as features. In the second data set, called *BioASQ labels*, the labels attached to each article were used as word tokens and thus the vocabulary of the corpus was equal to the set of labels. This simulates a scenario, where one would want to cluster the labels based on LDA.

4.1.2 Reuters-21578

The Reuters-21578 data set[1] has been widely used among researchers for almost two decades. It contains 21,578 documents from the Reuters news-wire. In our experiments we used the first 11,000 documents (the first 10,000 for training and the rest 1,000 for testing) removing

[1] `https://archive.ics.uci.edu/ml/datasets/Reuters-21578+Text+Categorization+Collection`

common stop-words and words appearing more than 1,000 times and less than 10 times in the corpus.

4.1.3 TASA subset

This data set is a small subset of 5,000 documents from a 37,000 document collection of diverse educational materials (e.g health, sciences, etc) collected by Touchstone Applied Science Associates (Landauer et al., 1998). The first 4,750 documents were used as a training set and the remaining 250 as a test set. The corpus already had stop-words and infrequent words removed, so we did not perform any further pre-processing.

4.2 Evaluation

It is difficult to find an objective approach to evaluate the prediction quality of a trained LDA model, as there is no underlying ground truth to compare predictions to, since it is an unsupervised task. Furthermore, the classic unsupervised learning and clustering evaluation measures cannot be used, as LDA does not assign an instance to a specific cluster (or topic) but rather makes a soft-assignment by associating a topics distribution with every instance. We considered two methods for evaluating the quality predictions generated by an unsupervised LDA model: (1) Perplexity evaluations and (2) Gibbs1 evaluations.

4.2.1 Perplexity

Evaluation of LDA models typically focuses on the probability of a set of held-out documents given an already trained model (Wallach et al., 2009). In this context, one must compute the model's posterior predictive likelihood of all words in the test set, given estimates of the topic parameters ϕ and the document-level mixture parameters θ.

The likelihood of a set of test documents D, given an already estimated model M. is given by (Heinrich, 2004) as:

$$Likelihood = \sum_{d=1}^{D} logp(d|M) = \sum_{d}^{D} \sum_{w}^{N_d} log(\phi_{k,w} \times \theta_{d,k}) \tag{8}$$

and the perplexity as

$$Perplexity = exp(-\frac{Likelihood}{\sum\limits_{d=1}^{D} N_d}) \tag{9}$$

where lower values of perplexity signify a better fitted model.

A common practice in the literature in order to compute the above likelihood is to run the CGS algorithm for a few iterations on the first half of each document, and then to compute the perplexity of the held-out data, based on the trained model's posterior predictive distribution over words (Asuncion et al., 2009). This is the approach we follow.

4.2.2 Gibbs1 evaluation procedure

An alternative means for model evaluation is given by Yao et al. (2009), namely the Gibbs1 method (previously discussed in Sect. 2). We will use this approach as a secondary means of model comparison. For clarification, we provide the exact procedure for this evaluation method:

1. Fit and save an LDA model, I, to the training data using CGS until convergence.
2. Add the test data to the training set.
3. Initialize randomly the topic variables only for the test data.
4. Run CGS again until convergence. Save the final model F.
5. Use the θ estimates of the test set obtained from F, as a *benchmark* for evaluation.
6. Perform inference for the test data with the method to be evaluated, using the initially saved model I. The difference from step 2 is that, here, the ϕ parameters are fixed.
7. Calculate the θ' estimates and compare them to the *benchmark* θ by employing a measure to calculate (dis)similarity of distributions.

The idea behind this procedure is that the topics in the initial and the final model will be approximately aligned, providing a means for comparing the two sets of topics. More specifically, the θ estimates of the test set obtained from step 4 are taken as the ground truth, as they are good approximations of the θ that would be obtained if the test data was included with the training data during model training.

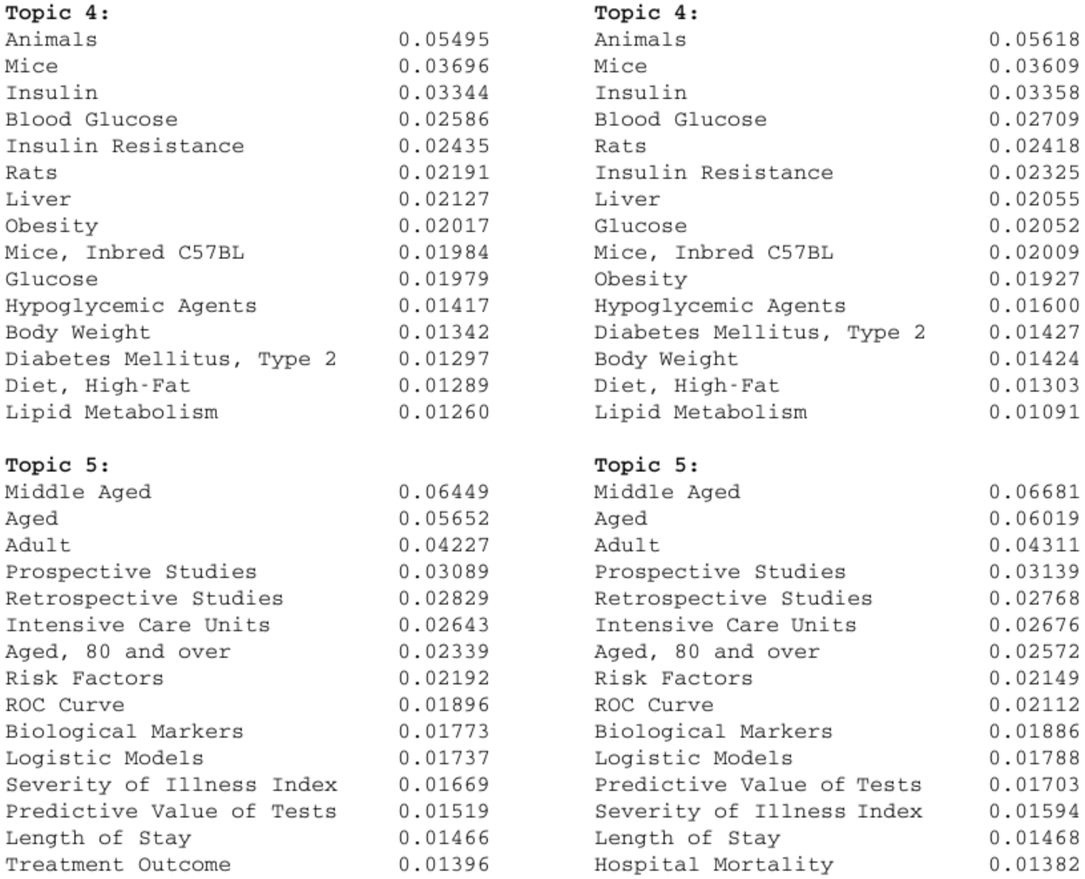
With this procedure, the model obtained during the initial estimation phase (model I) can be used to predict the test set's θ distributions, and if I and F are satisfyingly similar we can perform a comparison between the two different θ predictions. Fig. 1 shows an example of two topics in models I and F trained on the BioASQ labels corpus, where the topics have very similar ϕ distributions between the initial model I and the final model F. In order to ensure a minimal 'disruption' of the initially estimated model from the addition of the test data, we used only a small proportion of the data for testing (e.g. 1:20 for *BioASQ*). This procedure, being based on Gibbs sampling is not directly applicable to CVB0, hence we limited experiments to the four CGS methods in that case.

In order to measure the (dis)similarity between two distributions, following Yao et al. (2009) we used the following measures averaged over all test documents:

1. the JS-divergence. We used this measure instead of the KL-divergence that was initially proposed in (Yao et al., 2009) as the latter is defined on multinomial distributions only for non-zero values, an assumption that is violated in our case. On the contrary, the Jensen-Shannon divergence (Lin, 1991), which is a symmetric extension of the KL divergence, has the properties of accounting for zero values of the second distribution and is bounded by $log2$. It is defined as

$$JS(A,B) = \frac{1}{2}KL(A,M) + \frac{1}{2}KL(B,M)$$

in terms of the KL divergence, with $M = \frac{1}{2}(A+B)$.



```
Topic 4:                                        Topic 4:
Animals                         0.05495         Animals                         0.05618
Mice                            0.03696         Mice                            0.03609
Insulin                         0.03344         Insulin                         0.03358
Blood Glucose                   0.02586         Blood Glucose                   0.02709
Insulin Resistance              0.02435         Rats                            0.02418
Rats                            0.02191         Insulin Resistance              0.02325
Liver                           0.02127         Liver                           0.02055
Obesity                         0.02017         Glucose                         0.02052
Mice, Inbred C57BL              0.01984         Mice, Inbred C57BL              0.02009
Glucose                         0.01979         Obesity                         0.01927
Hypoglycemic Agents             0.01417         Hypoglycemic Agents             0.01600
Body Weight                     0.01342         Diabetes Mellitus, Type 2       0.01427
Diabetes Mellitus, Type 2       0.01297         Body Weight                     0.01424
Diet, High-Fat                  0.01289         Diet, High-Fat                  0.01303
Lipid Metabolism                0.01260         Lipid Metabolism                0.01091

Topic 5:                                        Topic 5:
Middle Aged                     0.06449         Middle Aged                     0.06681
Aged                            0.05652         Aged                            0.06019
Adult                           0.04227         Adult                           0.04311
Prospective Studies             0.03089         Prospective Studies             0.03139
Retrospective Studies           0.02829         Retrospective Studies           0.02768
Intensive Care Units            0.02643         Intensive Care Units            0.02676
Aged, 80 and over               0.02339         Aged, 80 and over               0.02572
Risk Factors                    0.02192         Risk Factors                    0.02149
ROC Curve                       0.01896         ROC Curve                       0.02112
Biological Markers              0.01773         Biological Markers              0.01886
Logistic Models                 0.01737         Logistic Models                 0.01788
Severity of Illness Index       0.01669         Predictive Value of Tests       0.01703
Predictive Value of Tests       0.01519         Severity of Illness Index       0.01594
Length of Stay                  0.01466         Length of Stay                  0.01468
Treatment Outcome               0.01396         Hospital Mortality              0.01382
```

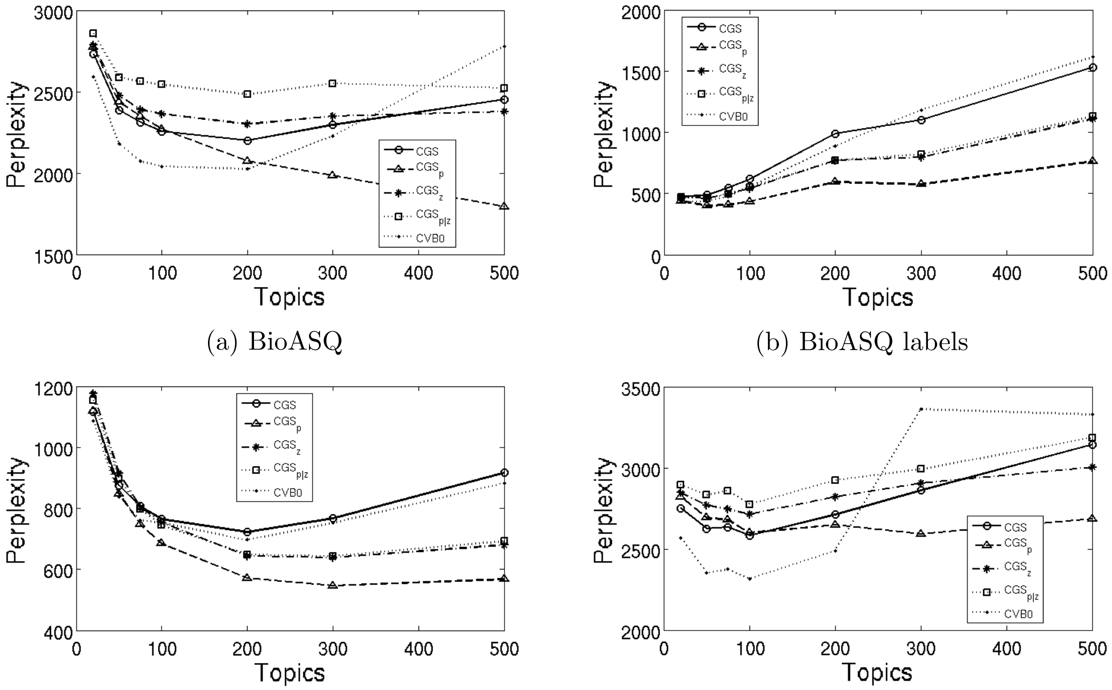
Figure 1: An example of the 15 most probable words for two topics for the BioASQ labels corpus. The topic distributions for the initial model are shown on the left while the ones for the final model on the right. The initial model is trained for 100 topics and 5000 iterations followed by the addition of the test data and 2500 iterations of Gibbs Sampling.

2. A document-pivoted macro-F1 score. For each document the n most probable topics are kept, where n is defined by requiring that the respective cumulative probability is 0.8, that is, $\sum_t \theta_t \leq 0.8$. These topics are then compared to the respective topics (i.e. that are retrieved in the same fashion) from the benchmark θ and the true positives, false positives and false negatives are calculated considering as ground truth the most probable topics of the benchmark θ.

4.3 Results

4.3.1 Perplexity Results

For this experiment we followed a similar approach to the one described in (Asuncion et al., 2009). During training we ran each chain for 500 iterations to obtain a single point estimate of the ϕ distributions. During prediction we ran 50 iterations from one chain for the first half of each document to obtain an estimate of θ_d. We then generated the posterior predictive



(a) BioASQ

(b) BioASQ labels

(c) Reuters

(d) TASA

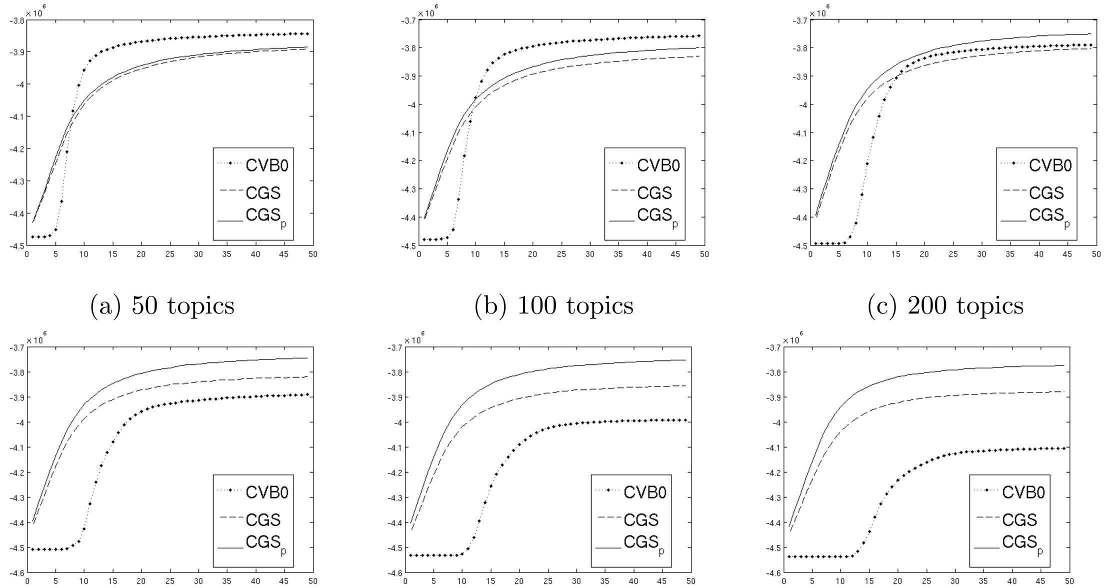
Figure 2: Perplexity against number of topics for the three CGS methods, standard CGS and CVB0. Results are taken by averaging over 5 different runs.

distributions from these estimates, and computed perplexity on the second half of each document. The α and β priors were fixed across all data sets both to 0.1. The above process was followed for both CGS and CVB0.

Figure 2 shows the perplexity results for all data sets across different settings (20, 50, 75, 100, 200, 300 and 500 numbers of topics) for all inference/prediction methods. There is a strong interaction between the perplexity scores, the data sets, and the number of topics. We believe that this is in part due to the diverse statistics, such as the average document length and the average number of features per document (ref. to Sect. 4.1), that characterize the data sets. For example, the performance generally improves on the BioASQ and Reuters data sets as the number of topics increases, while for the other two data sets smaller topic values appear to lead to better representations.

Despite the peculiarities of individual data sets, we can characterize some broad general trends in these results. First, CVB0 outperforms the other methods in two of the data sets when the number of topics is 200 or fewer. A possible explanation for this observation is related to the deterministic nature of CVB0; compared to its stochastic counterpart CGS, the algorithm is more prone to getting stuck in local maxima. As the number of topics increases, we expect the hypothesis space to grow bigger, making it more difficult for CVB0 to find a global optimum. CGS on the other hand, can exploit its stochastic nature to escape local maxima and converge to a better global representation of the data. Therefore, CVB0 could be better suited for cases where a small number of topics is required (in which case the fact that CVB0 converges a lot faster than CGS as shown by Asuncion et al. (2009) is an additional advantage), while CGS could fit better in the opposite case. A similar explanation may be the reason that CVB0 performs worse than most CGS methods across all topics on the Reuters and BioASQ labels dataset; these data sets have many fewer words per document, which may create a more difficult learning space with more local maxima.

Our proposed method CGS$_p$ outperformed standard CGS across nearly all experimental settings, and outperformed CVB0 in the majority of settings. More specifically, CGS$_p$ outperformed CVB0 in all settings, except with $L \leq 200$ on the BioASQ and TASA data sets. CGS$_p$ outperformed CGS except with $L \leq 100$ on TASA and $L \leq 100$ on BioASQ. The other two methods, CGS$_z$ and CGS$_{p|z}$ show a rather variable behavior, outperforming standard CGS and CVB0 in two out of the four data sets, and being consistently worse than CGS$_p$ in all cases.



(a) 50 topics (b) 100 topics (c) 200 topics

(d) 300 topics (e) 400 topics (f) 500 topics

Figure 3: CGS, CGS$_p$ and CVB0 convergence in terms of log-likelihood for a training of 50 iterations in the TASA subset for different values of topics. α and β were fixed to 0.1.

Another remark that could be made concerns the evolution of perplexity values as the number of topics rises. In all data sets we observe an over-fitting behavior (i.e. an increase in perplexity) after a specific number of topics. Interestingly, among all methods CGS$_p$ seems to have a smoother increase in perplexity values (hence a smoother 'deterioration' of the model), a fact that may provide an additional argument in that this method may be more successful in approximating the document - topics distributions.

4.3.2 CGS vs CVB0 in terms of convergence

The perplexity results of the previous section motivated us to further investigate the convergence behavior of CGS, CGS$_p$ and CVB0 algorithms. In particular, we wanted to

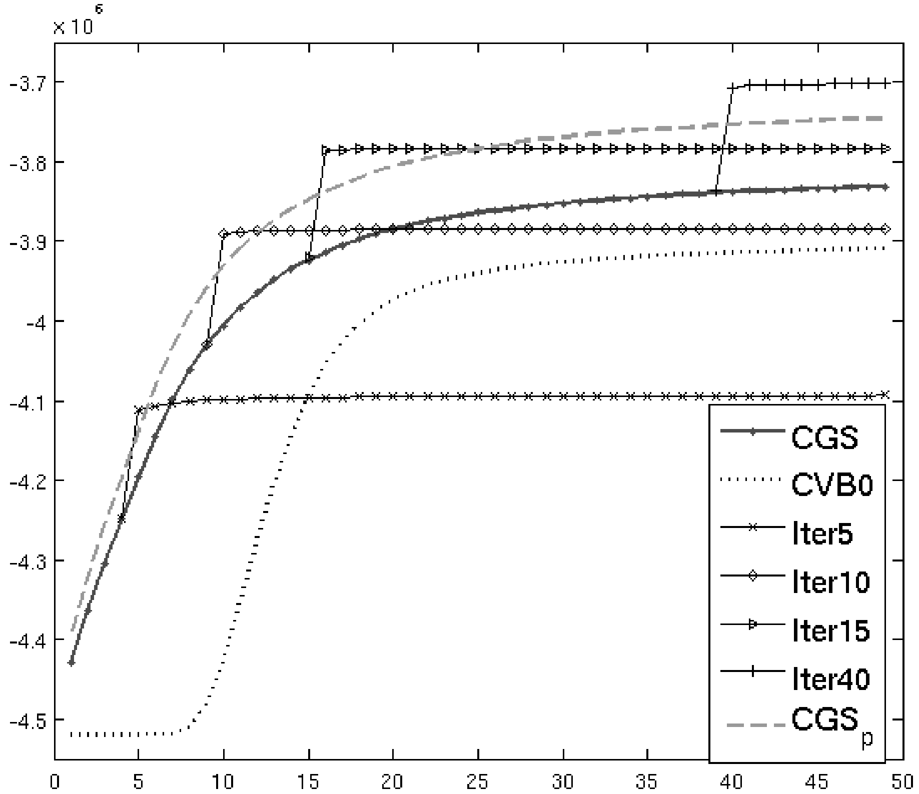


Figure 4: Convergence behavior for CGS, CGS$_p$, CVB0 and CGS-CVB0 blending, in terms of the Log-likelihood for a training of 50 iterations for the TASA subset. Iter n stands for passing from CGS estimation to CVB0 at iteration n. α and β values are fixed to 0.1.

know the conditions under which CGS (and our modifications of it) outperformed CVB0 and vice versa. Furthermore, we wanted to see how these differences evolved as the two algorithms converged upon their respective solutions.

Figure 3 presents a comparison of CVB0, CGS and CGS$_p$ (during the estimation phase) on the TASA dataset in terms of their log-likelihoods over iterations, under different numbers of topics. We note that, as CGS and CGS$_p$ use the same algorithm for estimation and inference, the only difference between them is that we substitute θ in Eq. 8 with θ_p of Eq. 6. The results seem to validate our previous observations; CVB0 is converging faster and to a higher log-likelihood than CGS and CGS$_p$ when $L \leq 100$. As the number of topics grows though, CGS (and CGS$_p$) performance catches up to CVB0 (around $L = 200$) and then clearly outperforms it at all iterations ($L \geq 300$). This supports our hypothesis that CVB0 performance is worse than CGS as the hypothesis space grows, due to it being a deterministic algorithm and getting stuck in local maxima. A secondary observation is that CGS$_p$ is converging steadily better than standard CGS, increasing its advantage as the number of topics grows.

We designed an additional experiment in order to further test the hypothesis that CGS-based methods are outperforming CVB0 with large topic numbers due to issues with local maxima. The basic idea behind this experiment was the following: if in fact CVB0 is performing worse under some conditions due to local maxima, then if the algorithm was initialized to a point lying outside that maxima, it would then ultimately converge to a better solution. Conversely, if CVB0 was performing worse under these conditions due to the fact that CVB0 is converging to an approximate solution (due to being a variational method), then it would converge to the same point independent of where it was initialized. Finally, since we know that CGS converges to better solutions than CVB0 under certain conditions, we can use these CGS solutions to initialize the CVB0 algorithm, and see what the resulting convergence behavior of CVB0 is.

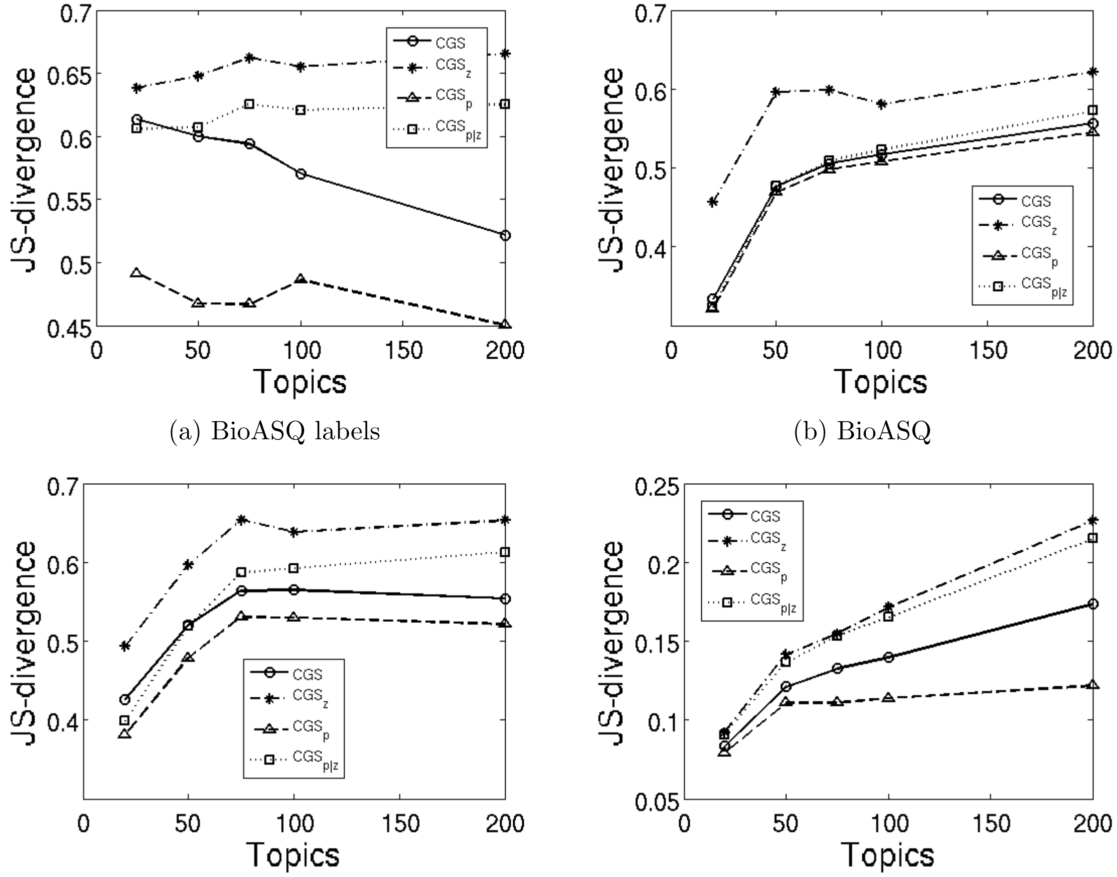
To examine this issue, we considered the case of $L = 300$ on the previously used $TASA$ corpus. We use CGS for training up to a number of iterations. Then, use the current CGS solution to initialize the CVB0 algorithm, and run CVB0 to completion. In order to pass the CGS solution from the Gibbs sampler to CVB0, we initialize the γ values of CVB0 to the current sampling distributions of CGS (Eq. 1), calculating the new n_w and n_d counts accordingly. In Figure 4 the convergence of CGS, CVB0 and four combinations of them (i.e., four time-points at which we switch from CGS to CVB0) are shown in terms of the likelihood of the models, during the estimation phase. CGS$_p$ is also included in order to further compare its convergence behavior against the rest of the methods. We can see that in all cases, as CVB0 takes over from the Gibbs Sampler, a very steep improvement is observed in likelihood values, followed by a rapid convergence of the CVB0 algorithm. A key result here is the following: when we transition from CGS to CVB0 at a point at which CGS has surpassed the baseline CVB0 solution in terms of likelihood, CVB0 converges to a clearly better solution than CVB0 does on its own (i.e., when only CVB0 is used). This strongly indicates that CVB0 is performing worse than CGS-based methods due to getting stuck in local maxima. Interestingly, when the transition from CGS to CVB0 is done before the CGS model likelihood surpasses that of CVB0-only, the CVB0 algorithm converges to a worse solution. This suggests that, in these cases, the CGS initialization actually moves the CVB0 model into a region with a worse local maxima.

These results seem to be consistent with the observations made by Teh et al. (2006) (refer to Sect. 4) and could eventually emerge from either (or both) of the following factors:

- Variational inference methods (VB, CVB and CVB0) approximate the true posterior by setting an upper bound to the negative log-likelihood (Blei et al., 2003; Teh et al., 2006; Asuncion et al., 2009). Specifically in the case of CVB0, as illustrated in (Foulds, 2014) Sect. 4.4.1, the algorithm is a result of three consecutive approximations of the initial problem. On the other hand, CGS is taking samples directly from the exact solution.

- CVB0 seems not to be able to avoid local maxima due to its deterministic nature, especially as the hypothesis space grows bigger, while CGS takes advantage of its stochastic nature in order to avoid them.

Finally, the above experimental setup suggests that a CGS-CVB0 hybrid approach could be more successful than the original CGS and CVB0 inference algorithms in estimating an LDA model's parameters, providing a possible future extension of this work.



(a) BioASQ labels

(b) BioASQ

(c) Reuters

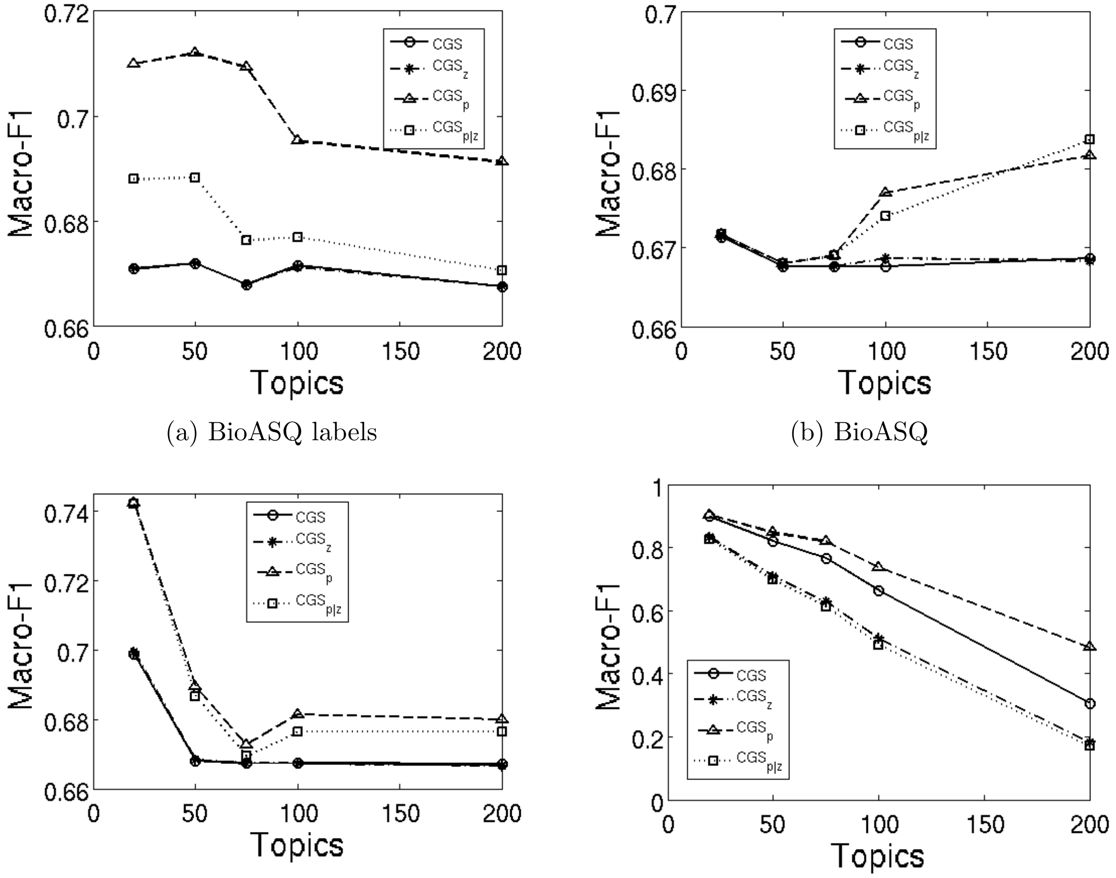
(d) TASA

Figure 5: JS-divergence against number of topics for the three CGS methods and standard CGS. Results are taken by averaging over 5 different runs.

4.3.3 JS-Divergence and F1-score results

In the second series of experiments the four CGS methods were compared by following the Gibbs1 procedure described in Section 4.2.2. We note again that we did not include CVB0 in these experiments, as this evaluation procedure is only directly applicable to Gibbs sampling algorithms.

In this experiment, we used more iterations than for the perplexity experiments, to ensure convergence for all procedures. During training we took a single sample after running Gibbs sampling for 5000 iterations, and used this sample to compute the ϕ parameters (corresponding to step 1 in Section 4.2.2). After the test documents were added (step 2, 3), the sampler was run for an additional 2600 iterations (step 4), and multiple samples were taken to compute the θ parameters of the test documents (step 5). During this phase we used a burn-in period of 100 iterations (to allow convergence) and a sampling lag of 5 iterations (to reduce auto-correlation). During prediction (step 6, 7), one Markov chain was ran for 2600 iterations, again with a burn-in period of 100 and a sampling lag of 5 iterations. Samples were taken for the four methods from the same chain in order to ensure that variations in their performance would not be affected by sampling from different states. The values of the α and β parameters were fixed across all data sets to $\frac{0.5}{L}$ and 0.01 respectively.



(a) BioASQ labels

(b) BioASQ

(c) Reuters

(d) TASA

Figure 6: Document-pivoted Macro-F1 measure against number of topics for the three CGS methods and standard CGS. Results are taken by averaging over 5 different runs.

In Figures 5 and 6 we present the results of the Gibbs1 experiments in terms of the Jensen-Shannon divergence and F1-score for the four data sets. These results are consistent with the perplexity results, in that they show a steady advantage of the CGS$_p$ predictions over standard CGS predictions, as well as the other two prediction methods. In the JS divergence plots, CGS$_p$ has similar values to CGS and CGS$_{p|z}$ in one case, but clearly outperforms all methods in the rest of the data sets. In terms of Macro-F1, CGS$_p$ again has an overall lead over the other methods, performing similar to CGS$_{p|z}$ in two cases and outperforming the rest of the methods on all data sets. Performance of the other methods is less consistent, with CGS_z and $\text{CGS}_{p|z}$ outperforming standard CGS prediction in some cases, while being inferior to it in others.

4.4 Discussion

In summary, our unsupervised learning experiments show a consistent advantage of our CGS_p method over the standard CGS prediction method. Furthermore, CGS_p tended to perform better than the predictions generated by CVB0. On two out of the four data sets tested, CGS_p outperformed CVB0 across all topic settings, and on the other two data sets CGS_p outperformed CVB0 when $L > 200$. The other two methods proposed, $\text{CGS}_{p|z}$ and CGS_z fail to improve over standard CGS in most cases, and do not seem to be useful approaches to follow.

It is perhaps unsurprising that CGS_p outperforms CGS, since the predictions use the same underlying information, except that CGS_p uses a richer representation of the θ parameters, since it stores the entire probability mass of the sampling distributions (similar to the CVB0 algorithm). More surprising is the fact that CGS_p outperforms CVB0. Based on our results, it seems that this is due to the fact that CVB0, being a deterministic algorithm, is more likely to get stuck in local maxima when the hypothesis space is sufficiently large, whereas CGS (and hence, our CGS_p method), is not as susceptible to this issue.

5. Multi-label learning experiments - Prior-LDA

Another important context for comparing different approaches to LDA prediction is in a multi-label, supervised setting. Here, we considered a multi-label learning extension of LDA—Prior-LDA—and used this as a basis for comparisons of model predictions. Following the same organization as the previous section, we present the data sets, the evaluation measures, the experimental setup, and lastly the results with a discussion of their implications.

5.1 Data sets

In this series of experiments we used four data sets: Delicious, BioASQ, EUR-Lex and NYT. Table 4 presents the relevant statistics. These data sets were chosen as representative of the diversity of real-world data, where there are often many labels, and sometimes not many features per training instance. Figure 7 depicts the frequency distribution of the labels for the data sets; these power-law like distributions are a characteristic feature of many real-world scenarios (Rubin et al., 2012).

5.1.1 Delicious

Delicious[2], contains textual data from web pages along with their tags, both extracted from the social bookmarking website of the same name (Tsoumakas et al., 2008). We did not perform any further pre-processing of this data set. A notable aspect of *Delicious* is that it contains very few features per instance both in the training and test sets, making accurate classification difficult.

[2] `http://mulan.sourceforge.net/datasets-mlc.html`

	Documents			Labels			
Data set	Training	Test	Average Length	Set	Cardinality	Average Frequency	Word Types
Delicious	12,910	3,181	16.72	983	19.06	250.34	500
BioASQ	20,000	10,000	113.55	16,311	13.85	16.98	22,394
NYT	14,668	7,000	581.13	4,185	5.45	19.10	24,307
EUR-Lex	10,000	9,314	986.01	3,532	5.40	18.34	21,168

Table 4: Statistics for the data sets used in the experiments. 'Label cardinality' stands for the average number of labels per document and 'label frequency' is the average label frequency. All figures concerning labels and word types are given for the respective training sets.

5.1.2 BioASQ

We used the same data set as for the unsupervised learning experiments and the same pre-processing procedure, the only difference being that, apart from the $20,000$ documents, we used an additional $10,000$ instances which served as test set.

5.1.3 New York Times corpus

This data set contains articles published by the New York Times and manually annotated via the New York Times Indexing Service. We used the same data set as in (Rubin et al., 2012), with the same size of training set (14,668 documents) and keeping the first 7,000 documents for testing (out of the 15,989 of the original paper). An additional difference from the experiments in (Rubin et al., 2012) is that during evaluation, the performance on all labels was used (not just the labels appearing in the test set) in order to be able to detect all false positive errors.

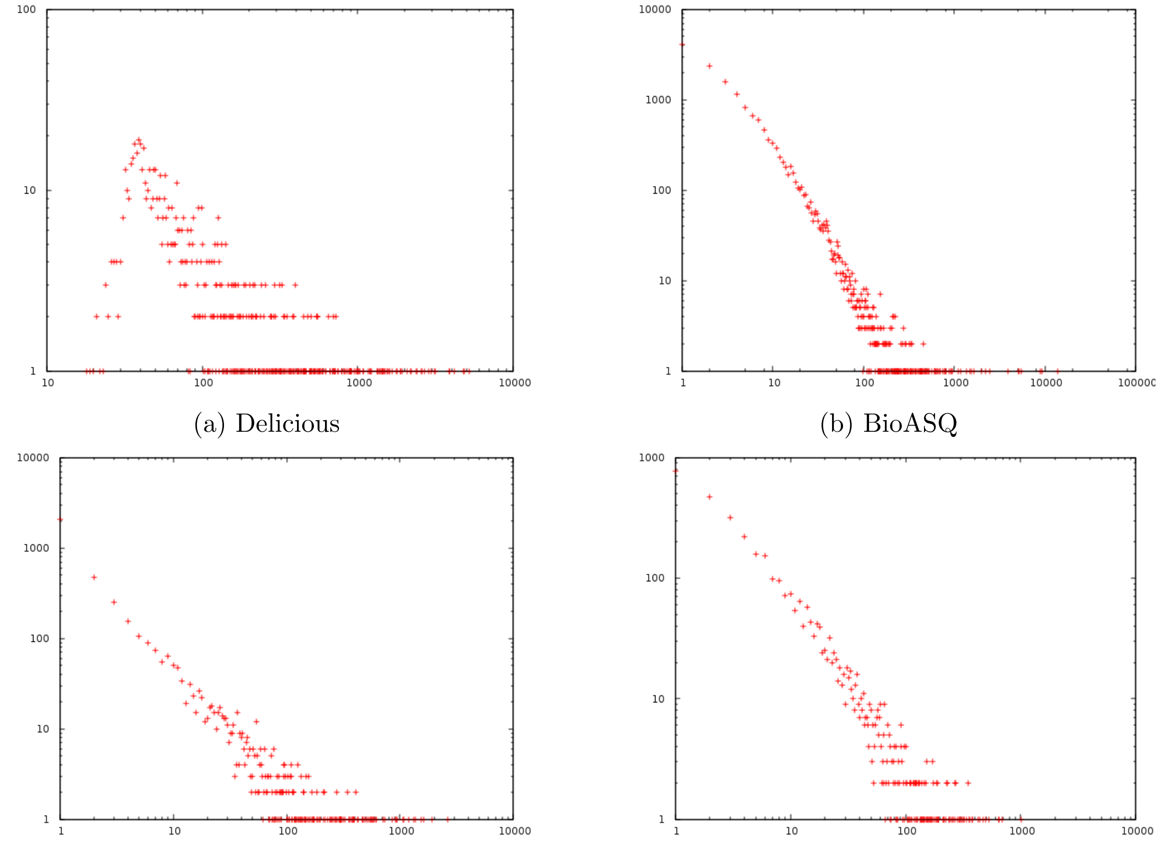
5.1.4 Eur-Lex corpus

The EUR-Lex dataset[3] consists of European Union legal documents (treaties, legislation, case-law and legislative proposals) (Loza Mencia and Fürnkranz, 2008). The downloaded corpus had a size of 19,347 documents. We kept the first 10,000 documents for training and the rest for testing. As the corpus was already stemmed and tokenized, the only pre-processing we did was to remove words with fewer than 5 occurrences. The set of labels contained 4,185 descriptors from EuroVoc, EU's multilingual thesaurus.

5.2 Evaluation metrics

We considered two widely-used performance measures for our multi-label experiments: the micro-averaged and macro-averaged F1 measures (Micro-F1 and Macro-F1, for short) (Tsoumakas et al., 2010). These measures are a weighted function of precision and recall, and emphasize the need for a model to perform well in terms of both of these underlying measures. The Macro-F1 score is the average of the F1-scores that are achieved across all

[3] `http://www.ke.tu-darmstadt.de/resources/eurlex`



(a) Delicious

(b) BioASQ

(c) NYT

(d) EUR-Lex

Figure 7: Labels distribution for the data sets used in the multi-label learning experiments. Both axes are in log-scale. Axis x represents the frequencies of labels (that is how many documents have this label in the corpus) and axis y represents the number of labels having frequency x. All data sets' distributions except for the first one, resemble to a power-law distribution.

labels, and the Micro-F1 score is the average F1 score weighted by each label's frequency. Therefore, Macro-F1 tends to emphasize performance on infrequent labels, while Micro-F1 tends to emphasize performance on frequent labels.

Given a multi-label classification problem with L labels, the Macro-F1 and Micro-F1 measures are defined in equations 10 and 11, in terms of the true positives (tp_l), false positives (fp_l) and false negatives (fn_l) of each label l.

$$Micro-F1_{score} = \frac{2 \times \sum\limits_{l=1}^{L} tp_l}{2 \times \sum\limits_{l=1}^{L} tp_l + \sum\limits_{l=1}^{L} fp_l + \sum\limits_{l=1}^{L} fn_l} \tag{10}$$

$$Macro - F1_{score} = \frac{1}{L}\sum_{l=1}^{L}\frac{2 \times tp_l}{2 \times tp_l + fp_l + fn_l} \tag{11}$$

5.3 Setup of Prior-LDA (CGS and CVB0)

The Prior-LDA model (Rubin et al., 2012) is an extension of the Labeled LDA algorithm (Ramage et al., 2009) that takes into account the relative label frequencies in the corpus, where test-documents are biased to assign words to more frequent labels. A non-symmetric Dirichlet prior α on the θ distributions is used at test-time to impose this bias. During training we kept the α parameter symmetrical and set it to:

$$\alpha = \frac{50}{L} \tag{12}$$

while during prediction we incorporated the label frequencies by setting it to:

$$\alpha = 50 \times \frac{f_l}{\sum f_l} + \frac{30}{L} \tag{13}$$

with f_l standing for the frequency of label l. The β prior was set to 0.01.

For all data sets, 30 chains were used during training, taking 30 point estimates of ϕ from each, with a burn-in period of 50 iterations and a sampling interval of 5 iterations. All samples from all chains were averaged to calculate the final ϕ distribution for each label.

In the prediction phase, 5 chains were used. For each Gibbs sampling chain, we took 20 samples (these same samples were used to compute predictions for the CGS, CGS$_z$, CGS$_p$ and CGS$_{p|z}$ methods). Again, we used a burn-in period of 50 iterations and a lag of 5 iterations between each sample. Once more, all samples from all chains were averaged to obtain the final document-label estimate θ for each method. These estimates of θ were computed from the exact same CGS samples for all four methods, in order to ensure fairness between all approaches.

For CVB0 we used the same setup as the one described above. However, as CVB0 is a deterministic algorithm that does not benefit greatly from averaging samples, we took the following additional steps: (a) during training we trained 30 chains, but each time with a different random order of the documents and a different random initialization and (b) during testing we took 5 chains using again different random initializations.

Finally, as the θ distributions define a ranking of labels for every instance (or more correctly a distribution of labels for an instance), in order to obtain a hard assignment of labels to documents, that is to apply a threshold to the above ranking, we used the Meta-Labeler approach (Tang et al., 2009). In particular, we used a linear regression model to predict the number of labels per instance. To train this model, the same feature space as for the LLDA models was employed and the LibLinear package (Fan et al., 2008) was used for the implementation.

5.3.1 Implementation Issues for CGS

The Labeled LDA algorithm has a major advantage over the widely used Binary Relevance approach in the multi-label setup; instead of treating each classification problem independently, all labels are learned simultaneously allowing for improved modeling of the label correlations and inter-dependencies. The above difference also allows the Labeled LDA algorithm to better determine the relevant features for labels that have very few positive examples, as illustrated in (Rubin et al., 2012) (refer to Sect. 1.2). This advantage nevertheless is closely connected to the difficulty of parallelizing LDA inference, which in large data scenarios is very demanding from a time perspective. More specifically, if N is the number of iterations of the Gibbs sampler, and D, L and V_d as denoted in Table 1, then the algorithm has a time complexity of $\mathcal{O}(N \times D \times V_d \times L)$. For example in a scenario with orders of magnitude, $N \sim 10^2$, $D \sim 10^4$, $V_d \sim 10^2$, $L \sim 10^3$ this leads to approximately 10^{11} operations. However in Labeled LDA, during training the time complexity is a lot lower as each word of a document can only be assigned to that document's labels. During prediction however, the time efficiency issues must be dealt. A first and simple solution is to split the data set to be predicted and infer the distributions of interest in parallel, addressing the D factor. Another approach we took is the method proposed by Porteous et al. (2008) which gives an exact inference of the parameters. Their approach, Fast-LDA, exploits the fact that the probability mass of the distribution p (Eq. 1) after some iterations concentrates on very few topics, many orders of magnitude smaller than the total number of topics. Therefore, by setting the proper bounds to the unnormalized probability of Eq. 1 one can avoid to calculate the probabilities for all topics, restricting calculations to only a few of them. We followed both of these ideas to deal with the time efficiency issues. Particularly for FastLDA, some minor changes are needed to account for the prediction phase as in the initial paper the authors concentrate on the estimation of the LDA model. Based on the generalized version of Hölder's inequality, they define an upper bound Z_0 for the sum of the conditional probabilities over topics for each word of each document as

$$Z_0 = \|\vec{a}\|_p \|\vec{b}\|_q \|\vec{c}\|_r \geq \sum_k \vec{a_k}\vec{b_k}\vec{c_k}$$

with

$$\vec{a} = [n_{d,1} + \alpha, ..., n_{d,L} + \alpha]$$

$$\vec{b} = [n_{w_i,1} + \beta, ..., n_{d,L} + \beta]$$

$$\vec{c} = [1/(n_{sum(1)} + V\beta), ..., 1/(n_{sum(L)} + V\beta)]$$

and requiring that $1/p + 1/q + 1/r = 1$.

During prediction, instead of $\vec{b}$ and $\vec{c}$ we have respectively $\vec{\phi}$, so in this case

$$Z_0 = \|\vec{a}\|_p \|\vec{\phi}\|_q$$

with a natural choice for p and q being $p = q = 2$.

5.4 Results and Discussion

Tables 5 and 6 show the Micro-F1 and Macro-F1 results respectively for all algorithms on the four data sets. We additionally show the average rank of each model, in terms of how it performs among the five models on average across the four data sets.

First we consider the results for when only a single chain was used. In terms of both the Macro-F1 and the Micro-F1 measures, the CGS$_p$ method performed best on three out of the

		Micro-F			
Algorithm	Delicious	BioASQ	EUR-Lex	NYT	Avg Rank
		1 MC			
CGS	0.27503	0.35228	0.13514	0.37811	4
CGS_z	0.20137	0.40789	0.15254	0.38173	3.75
CGS_p	**0.29209**	**0.42356**	0.16032	**0.41924**	1.25
$\text{CGS}_{p\|z}$	0.21299	0.41706	0.15590	0.41403	2.75
CVB0	0.27731	0.33638	**0.18965**	0.36863	3.25
		5 MC			
CGS	0.27738	0.40438	**0.22249**	0.43075	2.75
CGS_z	0.27176	0.44445	0.22222	0.44504	3
CGS_p	**0.29740**	0.44765	0.21632	**0.45662**	1.75
$\text{CGS}_{p\|z}$	0.26889	**0.45192**	0.21601	0.45513	3
CVB0	0.27731	0.33975	0.18967	0.36916	4.5

Table 5: Results for the label-pivoted Micro-F1 measure comparing the Prior LDA models. For the LDA models, MC stands for Markov chains. From each chain, 20 samples were taken.

four data sets with the exception of the EUR-Lex data set where CGS_p was outperformed by CVB0 in terms of Micro-F1 and by $\text{CGS}_{p|z}$ in terms of Macro-F1. With respect to the other two methods, $\text{CGS}_{p|z}$ is the second best performing method overall, while CGS_z does not seem to bring a significant improvement over standard CGS. Standard CGS and CVB0 appear to perform similarly with the results being roughly even for the two measures (each algorithm is better in two out of four data sets).

Model performance shifts even further in favor of the CGS-based predictions when we use multiple chains to make predictions. Whereas the CGS algorithm benefits from averaging samples—due to the fact that each sample is essentially an independent draw from the posterior distribution—CVB0 is deterministic, and achieves very small improvements (if any) since it tends to converge to the same specific maximum. When 5 chains are used, all CGS-based methods outperform CVB0 in all but one case (the single exception being that CVB0 outperforms $\text{CGS}_{p|z}$ on the *Delicious* data set using the Micro-F1 measure). The only data set for which CVB0 performed roughly as well as the CGS-based measures was the *Delicious* data set.

In order to better understand the above results, we should consider the fact that in all the above experiments the number of labels (or, equivalently, topics) is fairly large compared to the previous experiments with unsupervised LDA. *Delicious* has 983 labels, while the rest of the data sets have label sets in the order of equal or greater than 10^3 (BioASQ has $16,311$ labels). During training, the Prior-LDA algorithm constrains each word and each document to be assigned only to labels belonging to the document's label set, which presumably simplifies the task of approximating the ϕ distributions. On the other hand, during prediction Prior LDA reduces to standard LDA and therefore the problem of effectively searching through an enormous hypothesis space arises again. This is likely the reason for the significantly inferior performance of CVB0 compared to the CGS methods.

In terms of the CGS-based methods, the results are more mixed than for 1 MC. Nevertheless, CGS$_p$ again maintains the overall advantage for the Micro-F1 measure, while performing roughly the same with CGS$_{p|z}$ and outperforming the rest of the methods for Macro-F1. The alternative approaches have a more variable performance: CGS$_z$ predictions are worse or similar to standard CGS predictions for both measures, while CGS$_{p|z}$ has a roughly equivalent performance to CGS for Micro-F1 (outperforming it in two out of four data sets) and a slightly better for Macro-F1.

		Macro-F			
Algorithm	Delicious	BioASQ	EUR-Lex	NYT	Avg Rank
		1 MC			
CGS	0.04194	0.22608	0.16557	0.60858	3.75
CGS$_z$	0.07980	0.35097	0.15045	0.55708	3.75
CGS$_p$	**0.10273**	**0.37930**	0.18516	**0.61648**	1.25
CGS$_{p\|z}$	0.09155	0.37625	**0.18570**	0.61248	1.75
CVB0	0.04208	0.19102	0.16934	0.52631	4.5
		5 MC			
CGS	0.04217	0.37838	0.18411	**0.65017**	3
CGS$_z$	0.09812	**0.39584**	0.18177	0.64035	3
CGS$_p$	0.10378	0.39061	**0.19611**	0.64983	2
CGS$_{p\|z}$	**0.10671**	0.39532	0.19453	0.64691	2
CVB0	0.04208	0.19376	0.16956	0.52647	5

Table 6: Results for the label-pivoted macro-F measure.

Overall, the results seem to validate the findings of the unsupervised learning experiments; CGS$_p$ seems to yield a steady improvement over standard CGS and CVB0 for multi-label learning setups as well. The other two methods proposed along this paper do not seem to provide a convincing case against standard CGS.

6. Conclusions and Future Work

In this work we dealt with the problem of LDA inference on unseen data, and more specifically with how to improve the predictions generated by the Collapsed Gibbs Sampling (CGS) algorithm. We proposed three variations of the standard CGS θ equation and investigated their performance on multiple experimental scenarios, both for unsupervised LDA and one of its multi-label learning variants, Prior LDA. We also compared our methods against the CVB0 algorithm, investigating also briefly the convergence behavior of CGS and CVB0. The overall results suggest a steady advantage of one of the alternatives proposed over the standard method to compute θ as well as a steady improvement over the CVB0 algorithm.

In Sect. 4.3.2, in order to study the convergence behavior of CGS and CVB0, we looked at the convergence behavior of CVB0 after being initialized with CGS. The resulting method showed an improved performance with respect to the original algorithms, under certain conditions. These results suggest that CVB0 performs worse than CGS under some conditions due to it converging to a local maximum. We plan to further investigate the results of this experiment by performing a more extensive empirical comparison of this hybrid method to the original inference methods and by studying in a more elaborate manner the theoretical properties of possible hybrid approaches.

Acknowledgments

The authors would like to thank the BioASQ team for kindly providing the BioASQ corpus that was used throughout the experiments.

References

L. AlSumait, D. Barbara, and C. Domeniconi. On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking. In *Data Mining, 2008. ICDM '08. Eighth IEEE International Conference on*, pages 3–12, Dec 2008. doi: 10.1109/ICDM.2008.140.

Arthur Asuncion. Approximate mean field for dirichlet-based models. In *ICML Workshop on Topic Models*. Citeseer, 2010.

Arthur Asuncion, Max Welling, Padhraic Smyth, and Yee Whye Teh. On smoothing and inference for topic models. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 27–34, Arlington, Virginia, United States, 2009. AUAI Press. ISBN 978-0-9749039-5-8.

Georgios Balikas, Ioannis Partalas, Axel-Cyrille Ngonga Ngomo, Anastasia Krithara, and Georgios Paliouras. Results of the bioasq track of the question answering lab at CLEF 2014. In *Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15-18, 2014.*, pages 1181–1193, july 2014.

David M. Blei and Jon D. McAuliffe. Supervised topic models. In *NIPS*, 2007.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003. ISSN 1532-4435.

Liangliang Cao and Li Fei-Fei. Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct 2007. doi: 10.1109/ICCV.2007.4408965.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874, June 2008. ISSN 1532-4435.

James Foulds, Levi Boyles, Christopher DuBois, Padhraic Smyth, and Max Welling. Stochastic collapsed variational bayesian inference for latent dirichlet allocation. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 446–454, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2174-7. doi: 10.1145/2487575.2487697.

James Richard Foulds. *Latent Variable Modeling for Networks and Text: Algorithms, Models and Evaluation Techniques DISSERTATION*. PhD thesis, UNIVERSITY OF CALIFORNIA, IRVINE, 2014.

Andre Gohr, Alexander Hinneburg, Rene Schult, and Myra Spiliopoulou. Topic evolution in a stream of documents. In *In SDM*, pages 859–872, 2009.

T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl. 1):5228–5235, April 2004.

Gregor Heinrich. Parameter estimation for text analysis. Technical report, 2004.

Matthew Hoffman, Francis R. Bach, and David M. Blei. Online learning for latent dirichlet allocation. In J.D. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 856–864. Curran Associates, Inc., 2010a. URL `http://papers.nips.cc/paper/3902-online-learning-for-latent-dirichlet-allocation.pdf`.

Matthew Hoffman, David M. Blei, and Francis Bach. Online learning for latent dirichlet allocation. In *NIPS*, 2010b.

Thomas K. Landauer, Peter W. Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse Processes*, (25):259–284, 1998.

Ximing Li, Jihong OuYang, and Xiaotang Zhou. Supervised topic models for multi-label classification. *Neurocomputing*, 149:811–819, 2015. doi: 10.1016/j.neucom.2014.07.053.

Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37:145–151, 1991.

Eneldo Loza Mencia and Johannes Fürnkranz. Efficient pairwise multilabel classification for large scale problems in the legal domain. In *12th European Conference on Principles and Practice of Knowledge Discovery in Databases, PKDD 2008*, pages 50–65, Antwerp, Belgium, 2008.

David Newman, Arthur U Asuncion, Padhraic Smyth, and Max Welling. Distributed inference for latent dirichlet allocation. In *NIPS*, volume 20, pages 1081–1088, 2007.

Yannis Papanikolaou, Grigorios Tsoumakas, Manos Laliotis, Nikos Markantonatos, and Ioannis Vlahavas. Large-scale semantic indexing of biomedical papers via a statistical significance multi-label ensemble. Manuscript submitted for publication, 2015.

Ian Porteous, David Newman, Alexander Ihler, Arthur Asuncion, Padhraic Smyth, and Max Welling. Fast collapsed gibbs sampling for latent dirichlet allocation. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 569–577, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-193-4. doi: 10.1145/1401890.1401960.

Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 248–256, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-59-6.

Daniel Ramage, Christopher D. Manning, and Susan Dumais. Partially labeled topic models for interpretable text mining. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 457–465, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0813-7. doi: 10.1145/2020408.2020481.

Timothy N. Rubin, America Chambers, Padhraic Smyth, and Mark Steyvers. Statistical topic models for multi-label document classification. *Mach. Learn.*, 88(1-2):157–208, July 2012. ISSN 0885-6125. doi: 10.1007/s10994-011-5272-5.

Lei Tang, Suju Rajan, and Vijay K. Narayanan. Large scale multi-label classification via metalabeler. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 211–220, New York, NY, USA, 2009. ACM.

Yee Whye Teh, David Newman, and Max Welling. A collapsed variational bayesian inference algorithm for latent dirichlet allocation. In *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, pages 1353–1360, 2006.

G. Tsoumakas, I. Katakis, and I. Vlahavas. Effective and efficient multilabel classification in domains with large number of labels. In *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08)*, pages 30–44, 2008.

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Mining multi-label data. In Oded Maimon and Lior Rokach, editors, *Data Mining and Knowledge Discovery Handbook*, chapter 34, pages 667–685. Springer, 2nd edition, 2010.

Hanna M. Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1105–1112, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553515.

Limin Yao, David Mimno, and Andrew McCallum. Efficient methods for topic model inference on streaming document collections. In *KDD*, 2009.

Haizheng Zhang, Baojun Qiu, C.L. Giles, Henry C. Foley, and J. Yen. An lda-based community structure discovery approach for large-scale social networks. In *Intelligence and Security Informatics, 2007 IEEE*, pages 200–207, May 2007. doi: 10.1109/ISI.2007.379553.

Bin Zheng, David C McLean, and Xinghua Lu. Identifying biological concepts from a protein-related corpus with a probabilistic topic model. *BMC bioinformatics*, 7(1):58, 2006.

Jun Zhu, Amr Ahmed, and Eric P. Xing. Medlda: Maximum margin supervised topic models for regression and classification. In *ICML*, 2009.