Improving Performance of Topic Models by Variable Grouping

Evgeniy Bart
Palo Alto Research Center
Palo Alto, CA 94394
bart@parc.com

Abstract

Topic models have a wide range of applications, including modeling of text documents, images, user preferences, product rankings, and many others. However, learning optimal models may be difficult, especially for large problems. The reason is that inference techniques such as Gibbs sampling often converge to suboptimal models due to the abundance of local minima in large datasets.

In this paper, we propose a general method of improving the performance of topic models. The method, called 'grouping transform', works by introducing auxiliary variables which represent assignments of the original model tokens to groups. Using these auxiliary variables, it becomes possible to resample an entire group of tokens at a time. This allows the sampler to make larger state space moves. As a result, better models are learned and performance is improved. The proposed ideas are illustrated on several topic models and several text and image datasets. We show that the grouping transform significantly improves performance over standard models.

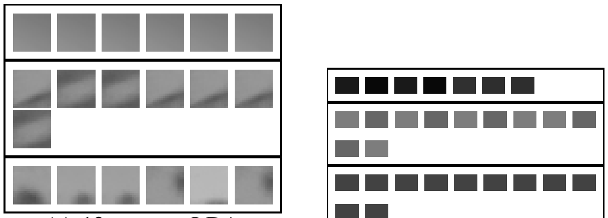
1 Introduction

Topic models such as Latent Dirichlet Allocation (LDA) are widely used in applications such as text modeling, collaborative filtering, and image segmentation [Blei *et al.*, 2003; Sivic *et al.*, 2005; Marlin, 2003; Andreetto *et al.*, 2007]. Since exact inference is intractable, iterative methods such as Gibbs sampling are typically used. Using these methods on large problems often produces poor models due to the abundance of local minima in large datasets.

In this paper, we propose a general method of improving the performance of topic models. The method works by combining the original topic variables into groups (as in Figures 1–2) in such a way that all variables within a group are likely to be assigned the same topic. Using these groups, it becomes possible to re-sample an entire group at a time (instead of a single variable at a time). This allows the sampler to make larger moves and thus converge to a better model.

The proposed method also touches upon a fascinating aspect of human problem solving—the human ability to discover the correct primitives at a level of abstraction suitable for a given problem. This is in contrast to most machine learning algorithms which only work with user-supplied primitives. For example, when Gibbs sampling is used for inference in a topic model, these primitives typically represent the most basic building blocks of the problem, such as image pixels. As the problem size increases, more variables in the model become necessary. Gibbs sampling often performs poorly under these circumstances, because standard primitives become too fine-grained for large problems. The proposed variable grouping method can be viewed as an approach to discover higher-level primitives for a given problem.

The remainder of this paper is organized as follows. In the next section, we briefly review the relevant previous work. In section 3, we describe the proposed variable grouping method in detail. Experimental results are presented in section 4. We conclude with general remarks in section 5.



(a) 13 scenes gLDA

(b) Corel 1000 gLDA

Figure 1: Example groups learned by gLDA on image data. Three groups per experiment are shown. For each group, the tokens it contains are displayed within a frame. For the 13 scenes dataset, the average image patch is shown for each vector-quantized SIFT descriptor.

has curves curve produced curve curves curve
the the the the the the the
layer layer propagation layer layer

Figure 2: Example groups learned by gLDA on text data (the NIPS articles corpus). Three groups are shown. For each group, the tokens it contains are listed within a frame.

2 Brief survey of previous work

Several existing modifications of LDA [Yao *et al.*, 2009; Porteous *et al.*, 2008b] focus on improving efficiency (speed and memory requirements) of inference by using efficient data structures and algorithms. However, these methods still produce regular LDA models. If LDA performs poorly on a given dataset, the same poor model will be produced, just more efficiently. In contrast, the method proposed here attempts to improve the model produced on a given dataset. Although speed and memory are not the focus of our method, note that many LDA speedup techniques can be applied to it to combine the advantages of both.

One method related to the grouping transform proposed here is 'block sampling' [Bishop, 2006]. In block sampling, several variables are selected and resampled as a group. One important difference from the grouping transform is that in block sampling, all possible combinations of values are considered. For example, if the block consists of K topic variables, each of which has T possible values, then a total of T^K values need to be considered. This makes block sampling computationally inefficient for large K. In contast, in the method proposed here all variables in a group are constrained to have the same value. In addition, it is unclear in general which variables should be selected to form a block. Therefore, standard block sampling is typically used when several variables in a model are known a priori to be related deterministically or almost deterministically. In contrast, the method proposed here allows learning which variables need to be grouped.

Another class of relevant approaches includes augmentation samplers and similar methods. For example, in [Swendsen and Wang, 1987], variables are grouped and sampled as a group to improve convergence. Similar ideas are used in [Barbu and Zhu, 2003; Tu, 2005]. A disadvantage is that finding a reasonable augmentation for a new problem may be difficult. The approach proposed in this paper is more general and applicable to a broader class of models. An additional difference is that traditional augmentation samplers generally do not try to find persistent groups. Instead, the groups change at every iteration (an iteration corresponds to resampling all variables once). This is in contrast to the approach proposed here, where groups are persistent across multiple iterations and are valuable by themselves. Similar comments pertain to split-merge methods (e. g. [Jain and Neal, 2000]).

Next, we review approaches that use persistent groups of variables. In [Ren and Malik, 2003] (in the context of image segmentation) individual image pixels are combined into groups, called 'superpixels', and subsequent processing is done in terms of these superpixels. Superpixels are defined by color similarity and proximity. A similar process is used in [Gomes *et al.*, 2008], using a more general definition of similarity. In general, these methods are application- or model-specific. In additon, groups in these methods have to be determined using properties of the objects the original variables represent (for example, color similarity of image pixels). Creating such groups could therefore be difficult if the objects do not have an obvious similarity measure (for example, in a recommendation system, how would one determine similarity between different users?). The grouping method proposed in this paper is more general, as it requires no separate definition of similarity, although it can use it if available.

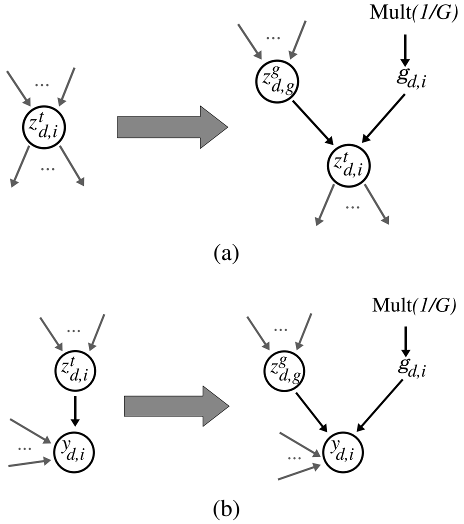


Figure 3: Grouping transform: a methodical way to introduce variable groups into topic models.

3 Variable grouping

In this section, we describe how variable grouping can be used to improve topic model performance. First, the proposed method of introducing group variables, called 'grouping transform', is described in section 3.1. An example application of grouping transform to LDA is described in detail in section 3.2. Additional applications of the grouping transform to two other topic models are described in sections 3.3, 3.4, and some properties of the grouping transform are described in section 3.5.

3.1 The grouping transform

In topic models, observations (called 'tokens') are assumed to be generated from a set of **topics**. Each topic represents a distinctive pattern of token co-occurence. For example, in LDA, topics are multinomial distributions (typically, sparse) over the vocabulary. For every token i in every document d, a topic model has a latent variable (called $z^t_{d,i}$) that represents the topic from which the token was generated.

The goal of training the topic model is to infer its parameters given a training set of documents. The most interesting of these parameters are usually the topics themselves (although others may be of interest too depending on the model). Collapsed Gibbs sampling is one of the most popular ways to train a topic model. Typically, continuous parameters (such as the topics) are integrated out, and topic assignments $z^t_{d,i}$ are sampled. The topics are recovered after sampling from these assignments.

Since even a short document may contain thousands of tokens, the total number of $z^t_{d,i}$'s may easily reach millions. As a result, sampling may exhibit poor convergence. Note, however, that the number of topics is relatively small compared to the number of tokens, so that many tokens belong to the same topic and have equal $z^t_{d,i}$'s. If several tokens could be identified in advance as belonging to the same topic, they could be resampled as a group.

We propose to replace the original topic model by a new model which represents groups of variables explicitly. This new model is obtained from the original model by a process called the 'grouping transform', described below.

First, we introduce a group assignment variable $g^t_{d,i}$ into the model. The idea is that document d has G_d groups. The variable $g^t_{d,i} \in [0, G_d - 1]$ then represents the group to which the token (d, i) (token i in document d) belongs. In the simplest case, these variables are generated from a uniform multinomial distribution.

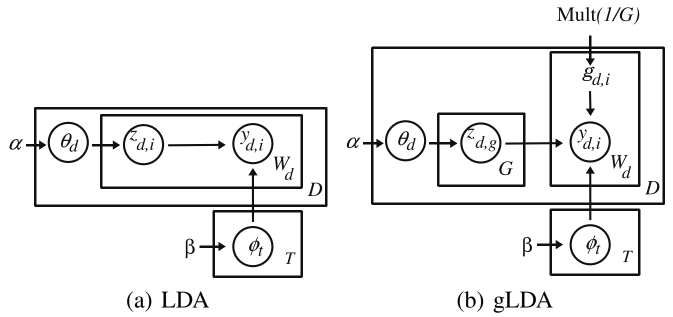
A group g in document d is assigned to a topic via the group topic variable $z^g_{d,g}$. (The superscript g indicates that one variable per group exists, while the superscript t indicates that one variable per token exists.) $z^g_{d,g}$ is sampled from the same distribution that $z^t_{d,i}$ was sampled from in the original model. An assignment $z^g_{d,g}$ means that all tokens in the group are generated by the topic $z^g_{d,g}$. Since the group for token (d, i) is $g^t_{d,i}$, the topic from which that token was generated is $z^g_{d,g_{d,i}}$. This process is illustrated in Figure 3(a).

Note that the proposed modifications change the model slightly. For example, in the original model the parents of the $z^t_{d,i}$ variables specified a distribution over N_d tokens in a document; in the modified model, they specify a distribution over the G_d groups. In other words, integrating out the new z^g and g^t variables will not produce the original model. This is in contrast to Swendsen-Wang-type approaches [Swendsen and Wang, 1987], where the same model is preserved. The main justification for the changes the grouping transform introduces is the improved performance compared to the original model.

The inference in the modified model can be performed by Gibbs sampling as well, similarly to the original model. One point to note is that each $z^t_{d,i}$ variable is a deterministic function of $g^t_{d,i}$ and $z^g_{d,g}$'s. As a result, the value of $z^g_{d,g}$ will never switch given fixed $z^t_{d,i}$'s; in other words, the sampler will not mix (as is typical with deterministic dependencies). To achieve mixing, we propose to sample each $z^g_{d,g}$ jointly with all the $z^t_{d,i}$ variables for tokens in group g, and to sample each $g^t_{d,i}$ jointly with the corresponding $z^t_{d,i}$. This does not add complexity to the sampler since in each case only one variable is actually sampled, while others are simply updated deterministically. Alternatively, the $z^t_{d,i}$ variables can be removed from the model completely; this is illustrated in Figure 3(b) for the case the observations are generated directly from the topics.

3.2 Group LDA

LDA is a popular model for text and image data [Blei *et al.*, 2003; Sivic *et al.*, 2005; Fei-Fei and Perona, 2005]. Its plate diagram is shown in Figure 4(a). LDA represents documents as bags of words. The i'th token in document d (denoted by $y_{d,i}$) is an instance of some word in the vocabulary. For text documents, the vocabulary is the set of English words. For images, each word is a cluster of visually similar image patches. A topic is a multinomial distribution (typically, sparse) over the vocabulary. These topics represent distinctive patterns of word co-occurence.



(a) LDA (b) gLDA

Figure 4: (a): the LDA model. (b): the gLDA model. In gLDA, a document d is generated as follows. First, θ_d, a mixture over the T topics is sampled from a uniform Dirichlet prior. A set of topic labels $z_{d,g}$ is generated for the G groups in the document. For each token, a group is sampled uniformly, and then a word is sampled from the topic associated to that group. The conditional distributions are: $\theta_d \sim \text{Dir}^T[\alpha]$, $\phi_t \sim \text{Dir}^Y[\beta]$, $g_{d,i} \sim \text{Mult}(1/G)$, $z_{d,g} \sim \text{Mult}(\theta_d)$, $y_{d,i} \sim \text{Mult}(\phi_{z_{d,g_{d,i}}})$.

The goal of training the LDA model is to infer model parameters given a set of documents. These model parameters include the topics ϕ_t, a topic distribution θ_d for each document d, and a topic assignment $z_{d,i}$ for each token in each document. Inference is typically performed by collapsed Gibbs sampling. The variables ϕ_t and θ_d are integrated out, and the z's are sampled according to:

$$p(z_{d,i} = z|\text{rest}) \propto \frac{\alpha + N^{\neg(d,i)}_{d,z}}{\alpha T + N^{\neg(d,i)}_{d,\cdot}} \frac{\beta + N^{\neg(d,i)}_{z,y_{d,i}}}{\beta Y + N^{\neg(d,i)}_{z,\cdot}}. \qquad (1)$$

Here N are the count variables: $N_{d,z}$ is the number of tokens in document d assigned to topic z, and $N_{z,y}$ is the number of times word y is assigned to topic z. The superscript $\neg(d, i)$ means that the i'th token in document d is omitted from the counts. A dot in place of a subscript represents summation over that subscript; for example, $N_{z,\cdot}$ is the total number of tokens assigned to topic z.

Applying the grouping transform described above to LDA, we obtain a model called gLDA (group LDA), shown in Figure 4(b). Note again that the meaning of θ^{gLDA}_d in gLDA is slightly different from that in LDA: in LDA, θ^{LDA}_d specifies the distribution of W_d tokens in a document, while in gLDA, θ^{gLDA}_d specifies the distribution of G groups. The gLDA model is nevertheless useful, because in many cases the parameters of interest are topics ϕ_k, whose meaning is not changed. If needed (e. g. for document classification), the counts $N^{\text{LDA}}_{d,z}$ can be computed after sampling (using $z^{\text{LDA}}_{d,i} = z_{d,g_{d,i}}$), and θ^{LDA}_d can be estimated from these.

Gibbs updates in gLDA are performed according to

$$p(g_{d,i}=g|\text{rest}) \propto \frac{\beta + N^{\neg(d,i)}_{z_{d,g},y_{d,i}}}{\beta Y + N^{\neg(d,i)}_{z_{d,g},\cdot}}, \qquad (2)$$

$$\begin{aligned} p(z_{d,g}=z|\text{rest}) \quad \propto & \quad \frac{\alpha + N^{\neg(d,g)}_{d,z}}{\alpha T + N^{\neg(d,g)}_{d,\cdot}} \cdot \\ & \cdot \frac{\Gamma(\beta Y + N^{\neg(d,g)}_{z,\cdot})}{\Gamma(\beta Y + N^{\neg(d,g)}_{z,\cdot} + N_{d,g})} \cdot \\ & \cdot \prod_y \frac{\Gamma(\beta + N^{\neg(d,g)}_{z,y} + N_{d,g,y})}{\Gamma(\beta + N^{\neg(d,g)}_{z,y})} . (3) \end{aligned}$$

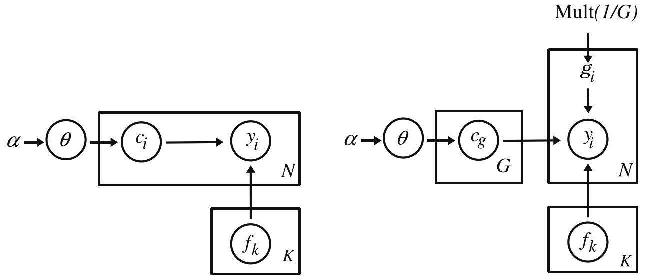
Here $N_{d,g}$ is the number of tokens in document d assigned to group g, and $N_{d,g,y}$ is the number of times word y is assigned to group g in document d. The superscript $\neg(d,g)$ means that all tokens in group g in document d are omitted from the counts. The remaining counts are as in LDA. The performance of gLDA is described in section 4.

Note that in eq. (2), the probability that token (d,i) is assigned to group g depends only on $z_{d,g}$ (the topic for group g). Typically, there are many groups in a document assigned to the same topic, and all these groups have equivalent probabilities. To improve convergence further, we assign the token (d,i) to that group which has the most tokens belonging to the same word $y_{d,i}$. In other words, group sampling is performed as follows. A token (d,i) is an instance of the word $y_0 = y_{d,i}$. First, the group g is sampled for this token according to eq. (2). The topic for this group is $z_0 = z_{d,g}$. Then, out of all groups with this topic z_0 we pick the one which has the largest number of other instances of the word y_0, and set $g_{d,i}$ to that group. This has the effect of encouraging groups to have uniform tokens, which improves convergence. Note that despite this, only about 30% of the groups in our experiments consist of instances of only one word; most groups consist of two words or more (see e. g. Figures 1–2). Note also that there are multiple groups per document and different instances of the same word may belong to more than one group. Therefore, dealing with polysemy is still possible.

3.3 Group AZP

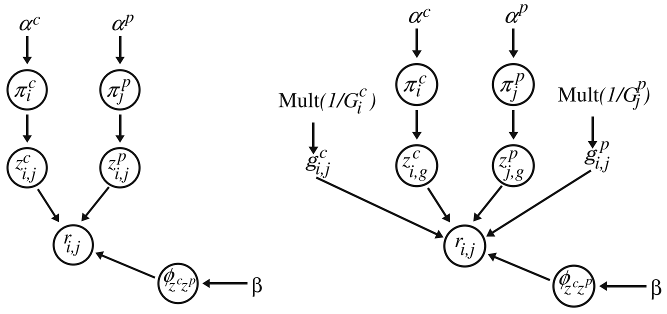
A topic model for probabilistic image segmentation was introduced in [Andreetto *et al.*, 2007]. This model will be called AZP here. Its plate diagram is shown in Figure 5(a). Image pixels in this model are generated from a mixture of topics, each topic being modeled as a mixture of a parametric component (a Gaussian distribution over colors) and a non-parametric component. See [Andreetto *et al.*, 2007] for details. For inference, θ is integrated out and the c_i variables are sampled according to

$$p(c_i = c|\text{rest}) \propto \frac{\alpha + N_c}{\alpha K + N} f_c(y_i). \qquad (4)$$



(a) AZP (b) gAZP

Figure 5: (a): the AZP model for image segmentation. An image is generated as follows. First, θ, a mixture over the K clusters (segments) is sampled from a uniform Dirichlet prior. A cluster label c_i is generated for every pixel in the image. A color is sampled from the 'topic' (a distribution f_{c_i} over colors) associated to that cluster. The cluster distributions f_k are mixtures of a parametric (Gaussian) and a non-parametric distributions. (b): the gAZP model. Each pixel is assigned to one of G groups. The remaining model is similar to AZP.



(a) BiLDA (b) gBiLDA

Figure 6: (a): the BiLDA model. The conditional distributions are: $\pi_i^c \sim \text{Dir}[\alpha^c]$, $\pi_j^p \sim \text{Dir}[\alpha^p]$, $\phi_{z^c,z^p} \sim \text{Dir}[\beta]$, $z^c_{i,j} \sim \text{Mult}(\pi^c_i)$, $z^p_{i,j} \sim \text{Mult}(\pi^p_j)$, $r_{i,j} \sim \text{Mult}(\phi_{z^c_{i,j},z^p_{i,j}})$. (b): the gBiLDA model. The conditional distributions are: $\pi_i^c \sim \text{Dir}[\alpha^c]$, $\pi_j^p \sim \text{Dir}[\alpha^p]$, $\phi_{z^c,z^p} \sim \text{Dir}[\beta]$, $g^c_{i,j} \sim \text{Mult}(1/G^c_i)$, $g^p_{i,j} \sim \text{Mult}(1/G^p_j)$, $z^c_{i,g} \sim \text{Mult}(\pi^c_i)$, $z^p_{j,g} \sim \text{Mult}(\pi^p_j)$, $r_{i,j} \sim \text{Mult}(\phi_{z^c_{i,g^c_{i,j}},z^p_{j,g^p_{i,j}}})$.

Group variables were introduced into AZP using the grouping transform. The resulting model (called gAZP) is shown in Figure 5(b). The inference is performed according to

$$\begin{aligned} p(c_g = c|\text{rest}) \quad & \propto \quad \frac{\alpha + N_c}{\alpha K + G} \cdot \prod_{i \text{ in group } g} f_c(y_i), \qquad (5) \\ p(g_i = g|\text{rest}) \quad & \propto \quad f_{c_g}(y_i). \qquad (6) \end{aligned}$$

Experiments with this model are presented in section 4.

3.4 Group BiLDA

A topic model for collaborative filtering was proposed in [Porteous *et al.*, 2008a; Airoldi *et al.*, 2008; Marlin, 2003]. This model will be called BiLDA here. It is assumed that N

Dataset	Method	Initial $\log p(\mathcal{W}, \mathcal{Z})$	Final $\log p(\mathcal{W}, \mathcal{Z})$	% improvement over LDA
NIPS stemmed	LDA	-7.31×10^6	-4.77×10^6	
	gLDA		-4.24×10^6	21%
NIPS raw	LDA	-7.12×10^6	-4.68×10^6	
	gLDA		-4.09×10^6	24%
NYT stemmed	LDA	-5.64×10^7	-3.60×10^7	
	gLDA		-3.42×10^7	9%
NYT raw	LDA	-8.97×10^7	-6.16×10^7	
	gLDA		-5.59×10^7	20%
Corel 1000	LDA	-8.20×10^6	-3.43×10^6	
	gLDA		-3.17×10^6	5%
13 scenes	LDA	-1.27×10^7	-8.14×10^6	
	gLDA		-7.28×10^6	19%

Table 1: Performance (in terms of $\log p(\mathcal{W}, \mathcal{Z})$) of LDA and gLDA. Higher values are better (note that the values are negative). Improvement is shown in percent relative to the difference between initial and final $\log p(\mathcal{W}, \mathcal{Z})$ of LDA. Higher improvement percentage indicates better performance.

customers have rated M products using R discrete rating values. For example, N binary images may consist of M pixels each, and each pixel's value is 0 or 1. In some cases, the data may be sparse. This is the case, for example, when some pixel values are missing (unknown) in some images. A typical task in this case is to predict the missing values.

Briefly, the approach in BiLDA is to model customer preferences by a set of customer topics, and product features by product topics. The distribution of ratings for a given customer topic and a given product topic is assumed to be a multinomial over the R values. The plate diagram for BiLDA is shown in Figure 6(a). As can be seen, it consists of two LDA models, one representing customer topics and another representing product topics. See [Porteous *et al.*, 2008a; Airoldi *et al.*, 2008] for details.

Group variables were introduced into BiLDA using the grouping transform. The resulting model (called gBiLDA) is shown in Figure 6(b). Equations for sampling in BiLDA and gBiLDA are omitted to save space, but can be derived in a straightforward manner. Experiments with BiLDA and gBiLDA are presented in section 4.

3.5 Properties of the grouping transform

Greediness of grouping A feature which is common to all the grouped models presented above is greediness of group formation. This refers to the fact that the probability of a token belonging to a group is proportional to the likelihood of the token under that group's topic. In contrast, in the original models this likelihood is gated by what can be thought of as the prior probability that the token belongs to that topic. For example, in LDA the probability of a token (d, i) belonging to a topic z is the likelihood of that token in the topic z (the second term in eq. (1)), gated by the probability of any token in document d belonging to that topic (the first term). In contrast, in gLDA only the analogue of the second term plays a role when resampling g (eq. (2)). Similarly, in AZP the likelihood of a pixel in a cluster (the f term in eq. (4)) is gated by the probability of that cluster in the image (the first term in eq. (4)). In contrast, only the f term plays a role in sampling the group assignments in gAZP (eq. (6)). Similar comments pertain to gBiLDA.

Selecting the number of groups A practical question is how should the number of groups be selected. Experimentally, we found that the performance is best with many groups per document—as many as tokens per document or more. Note that in this case many groups still contain more than one token, and other groups are empty. The nonempty groups contain 2–5 tokens on average. Using that many groups is, however, less efficient computationally. The performance is close to optimal when the number of groups is such that there are 3–10 tokens per group on average, and this also avoids the need to maintain and sample many empty groups. When the number of groups decreases further, the performance starts to deteriorate. Therefore, in our experiments, we set the number of groups such that there were on average 4 tokens per group. In the future, we plan to explore models where the number of groups is determined automatically using models similar to HDP [Teh *et al.*, 2006].

4 Results

Here, we evaluate the quality of models learned using group variables.

4.1 LDA and gLDA

Two text datasets and two image datasets were used for the experiments. For text experiments, we used the NIPS papers dataset [Roweis, 2002] and the New York Times corpus [Sandhaus, 2008] (only documents with at least 3500 words were used from the NYT corpus). Typically, text datasets are preprocessed by stemming the words and removing stop words. These versions of the datasets are called 'stemmed'. To illustrate how the models cope with raw data, we have also experimented with the unprocessed corpora (where neither stemming nor stop word filtering were performed). These versions are called 'raw'. In addition, the dataset of 13 visual scene categories [Fei-Fei and Perona, 2005] was used (we used vector-quantized SIFT descriptors as words, as

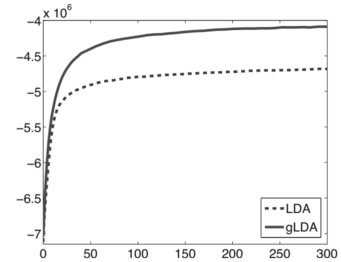


Figure 7: $\log p(\mathcal{W}, \mathcal{Z})$ as a function of iteration on the NIPS dataset. Top curve (solid green): gLDA. Bottom curve (dashed blue): LDA. For gLDA, the X axis shows the effective number of iterations that takes into account the fact that sampling one group variable requires more computation than in LDA.

described in [Fei-Fei and Perona, 2005]). Finally, an image dataset consisting of 1000 color images from the Corel dataset was used (we used color values, vector-quantized uniformly in the RGB space, as words).

The joint probability of all words $\mathcal{W}$ and all topic assignments $\mathcal{Z}$, $\log p(\mathcal{W}, \mathcal{Z})$, was recorded every 20 iterations and used as performance estimate [Wallach *et al.*, 2009]. In Figure 7, the performance as a function of iteration is plotted for regular LDA and gLDA. As can be seen, the gLDA model converges to a noticeably better mode compared to LDA. Performance of gLDA on additional datasets is summarized in Table 1. As can be seen, gLDA consistently outperforms LDA. Each experiment was repeated at least 10 times. All improvement figures are statistically significant (t test, $p << 0.001$).

Hyperparameter settings only weakly influence the performance of both methods. We used the settings recommended in [Steyvers and Griffiths, 2005]. Variations over two orders of magnitude affected the results by only 1–2 percentage points.

Block sampling is a standard method related to the grouping transform proposed here. As mentioned in section 2, for blocks of size K and with T topics, a total of T^K computations need to be performed to resample one block. This amounts to T^K/K computations per variable, as opposed to just T computations per variable for regular LDA. In our experiments we used $T = 100$ topics; as a result, block sampling was computationally infeasible for $K > 2$. We have run block sampling with blocks of two variables, but the improvement in performance was less than 0.5% compared to regular LDA. In contrast, gLDA often gives improvements of 10–20% (Table 1).

Several examples of learned topics are shown in Figure 8. As can be seen, gLDA copes much better with distractors (including stop words such as ‘the’), and as a result produces much better topics.

Several groups learned by gLDA on the NIPS dataset are shown in Figure 2. As can be seen, groups consist of tokens that often belong to the same topics. Most groups contain multiple words, although often the same word is repeated multiple times. Examples of groups learned on the image datasets are shown in Figure 1. Note that no measures of similarity between words were used. When more than one word is assigned to the same group, this assignment is based solely on the underlying model (gLDA). Nevertheless, the tokens in the same group usually contain similar words. For example, the middle group in Figure 1(a) contains various diagonal edges; the bottom group in the same figure contains different versions of blobs; the middle group in Figure 1(b) contains different shades of pink; and the topmost group in Figure 2 contains singular and plural versions of the word ‘curve’. Such groups, once established, could potentially be useful for other applications, such as training a different model on the same corpus.

the	training	neurons
order	the	neuron
of	and	the
mean	set	connections
field	et	of
approximation	generalization	network
and	validation	and
for	with	to
in	trained	their
theory	al	lateral

(a) Regular LDA

control	set	network
policy	and	networks
actions	test	neural
value	of	architecture
action	on	number
reward	validation	feed
controller	training	forward
optimal	is	chosen
function	error	sigmoidal
reinforcement	sets	reference

(b) gLDA

algorithm	have	network
only	algorithms	different
linear	field	simulation
step	sets	bounds
under	determined	curves
obtained	tasks	sub
log	experimental	clusters
energy	estimates	grid
reinforcement	entropy	demonstrated
computation	dynamical	free

(c) Restricted LDA

Figure 8: Example topics learned by LDA, gLDA and restricted LDA (see text) on the NIPS dataset. Three topics are shown for each model. For each topic, the 10 most probable words are shown. Note that gLDA topics are more focused and include fewer generic words (such as ‘the’, ‘and’, etc.). Note also that restricted LDA topics are much less coherent than both LDA and gLDA.

Since many groups in gLDA contain repetitions of the same word (Figures 1, 2), it was interesting to test a variant of LDA in which all instances of the same word were restricted to belong to the same topic. However, this restricted LDA performed very poorly in practice (much poorer than LDA). Examples of topics learned by restricted LDA are shown in Figure 8. The conclusion is that naively grouping all instances of the same word together does not improve performance.

4.2 AZP and gAZP

The performance of AZP was evaluated using the joint probability of all pixels $\mathcal{Y}$ and all cluster assignments $\mathcal{C}$ in the image, $\log p(\mathcal{Y}, \mathcal{C})$. The algorithms were tested on the dataset of images of egrets used in [Andreetto *et al.*, 2007], as well as on several personal photographs (mostly of landscapes). The improvement in performance of gAZP over AZP is defined in percent relative to the difference between initial and final $\log p(\mathcal{Y}, \mathcal{C})$ of AZP. On average, the performance improvement was 4.6%.

4.3 BiLDA and gBiLDA

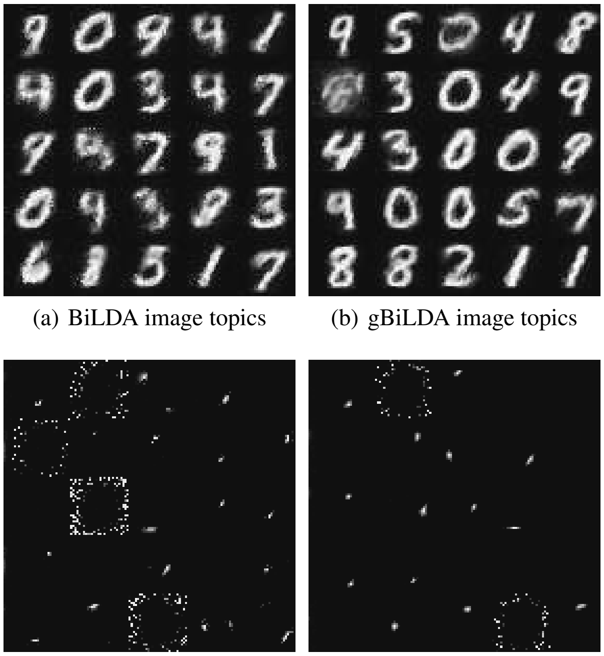
The MNIST dataset of handwritten digits [LeCun *et al.*, 1998] was used for the experiments. This dataset contains binary images of size 28×28 pixels. Some examples are shown in Figure 10(a). Each image corresponded to a customer in the BiLDA model, and each pixel corresponded to a product. Since the images are binary, $R = 2$ ratings were used. In each image, 50% of the pixels (selected at random) were observed, and the remaining 50% created a hold-out set. The BiLDA model with 100 customer, or image, topics and 100 product, or pixel, topics was fitted to the data by running Gibbs sampling for 1000 iterations. The gBiLDA model with the same parameters as BiLDA was fitted to the same data.

We visualized the image topics learned by the two models as follows. For every image topic z^c, a synthetic image of size 28×28 pixels was generated. For each pixel j, the probability of that pixel being white under z^c is displayed. This probability is computed as

$$\sum_{z^p} \pi_j^p[z^p] \cdot \mathrm{Mult}(1|\phi_{z^c,z^p}). \tag{7}$$

Several image topics for BiLDA are shown in Figure 9(a), and the image topics for gBiLDA are shown in Figure 9(b). As can be seen, each image topic represents a particular style of a digit. For example, several styles of the digit ‘0’, with varying aspect ratios and slants, are observed.

Several pixel topics learned by the two models are displayed in Figures 9(c), 9(d). For every pixel topic z^p, a synthetic image of size 28×28 pixels was generated. For each pixel j, the probability of this pixel being assigned to topic z^p is shown. This probability is given by $\pi_j^p[z^p]$, the z^p'th component of π_j^p. Referring to eq. (7), we may observe that to form a single image topic, the pixel topics are combined with weights given by $Mult(1|\phi_{z^c,z^p})$. The pixel topics thus represent *strokes*, which are subsequently combined to obtain complete characters. Several pixel topics represent the black border around the image, where there are rarely any white pixels. As can be seen, the gBiLDA strokes are more concentrated, compared to BiLDA's fuzzier strokes (this figure is best viewed on-screen).


(a) BiLDA image topics (b) gBiLDA image topics

(c) BiLDA pixel topics (d) gBiLDA pixel topics

Figure 9: Several image and pixel topics learned by BiLDA and gBiLDA. This figure is best viewed on-screen. See text for details.

The joint probability of all ratings $\mathcal{R}$ and all topic assignments $\mathcal{Z}$, $\log p(\mathcal{R}, \mathcal{Z})$, was recorded every 20 iterations and used as a quantitative performance estimate (cf. [Wallach *et al.*, 2009]). For BiLDA, $\log p(\mathcal{R}, \mathcal{Z})$ after initialization was -1.71×10^7, and converged to -1.97×10^6 after sampling for 1000 iterations. For gBiLDA, the performance converged to -1.67×10^6 (note that the values are negative and that higher numbers represent better performance).

As an additional performance measure we attempted to predict the missing pixel values and in this manner to reconstruct the images. For a pixel j in image i, the probability that this pixel is white was computed as

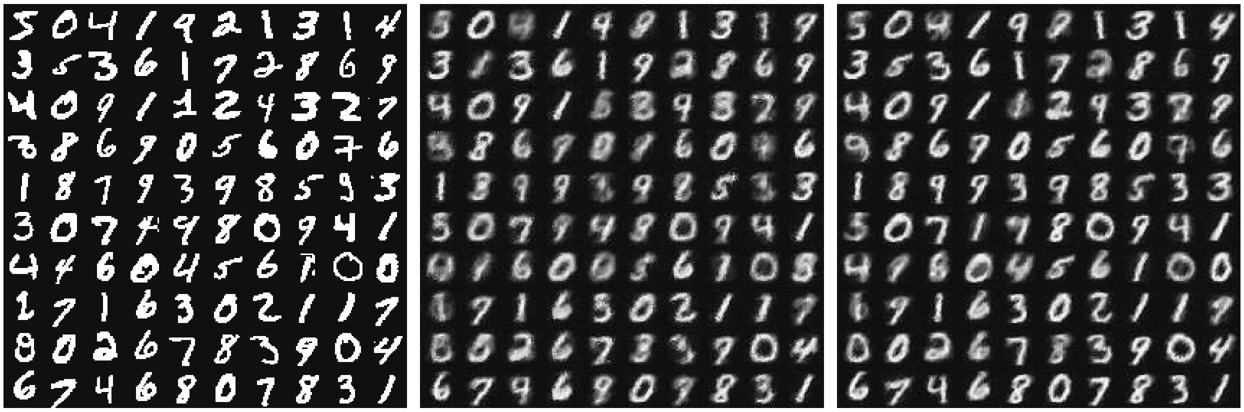
$$\sum_{z^c} \sum_{z^p} \pi_i^c[z^c] \cdot \pi_j^p[z^p] \cdot \mathrm{Mult}(1|\phi_{z^c,z^p}). \tag{8}$$

The reconstructions of several images obtained by BiLDA and gBiLDA are shown in Figures 10(b), 10(c). As can be seen, gBiLDA reconstruction quality is significantly better.

5 Discussion

We have presented a method for variable grouping that significantly improves topic model performance. The method was illustrated on three topic models and several text and image datasets.

An interesting problem for future research is introducing auxiliary similarity measures for individual words (e. g. edit



(a) Original images (b) BiLDA reconstructions (c) gBiLDA reconstructions

Figure 10: Left: several example images from the MNIST dataset. Middle: reconstruction of these images by BiLDA. Right: reconstruction of the same images by gBiLDA. As can be seen, gBiLDA reconstruction is much closer to the original.

distance for text). The framework provided by the grouping transform allows introducing such auxiliary information easily, by simply modifying the prior on the group variables. Another direction for future research is applying variable grouping to a broader class of graphical models.

References

[Airoldi *et al.*, 2008] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *JMLR*, 2008.

[Andreetto *et al.*, 2007] M. Andreetto, L. Zelnik-Manor, and P. Perona. Non-parametric probabilistic image segmentation. In *ICCV*, 2007.

[Barbu and Zhu, 2003] Adrian Barbu and Song-Chun Zhu. Graph partition by Swendsen-Wang cuts. In *ICCV*, 2003.

[Bishop, 2006] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[Blei *et al.*, 2003] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[Fei-Fei and Perona, 2005] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005.

[Gomes *et al.*, 2008] R. Gomes, M. Welling, and P. Perona. Memory bounded inference in topic models. In *ICML*, 2008.

[Jain and Neal, 2000] S. Jain and R. Neal. A split-merge Markov Chain Monte Carlo procedure for the Dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, 2000.

[LeCun *et al.*, 1998] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998.

[Marlin, 2003] B. Marlin. Modeling user rating profiles for collaborative filtering. In *NIPS*, 2003.

[Porteous *et al.*, 2008a] I. Porteous, E. Bart, and M. Welling. Multi-HDP: A non parametric bayesian model for tensor factorization. In *AAAI*, 2008.

[Porteous *et al.*, 2008b] I. Porteous, D. Newman, A. Ihler, A. Asuncion, P. Smyth, and M. Welling. Fast collapsed gibbs sampling for latent dirichlet allocation. In *KDD*, 2008.

[Ren and Malik, 2003] X. Ren and J. Malik. Learning a classification model for segmentation. In *ICCV*, 2003.

[Roweis, 2002] S. Roweis. Nips 12 dataset, 2002. www.cs.toronto.edu/˜roweis/data/nips12_str602.readme.

[Sandhaus, 2008] Evan Sandhaus. The new york times annotated corpus, 2008. Linguistic Data Consortium, PA.

[Sivic *et al.*, 2005] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering objects and their localization in images. In *ICCV*, pages 370–377, 2005.

[Steyvers and Griffiths, 2005] M. Steyvers and T. Griffiths. Probabilistic topic models. In T. Landauer, D. McNamara, S. Dennis, and W. Kintsch, editors, *Latent Semantic Analysis: A Road to Meaning*. Laurence Erlbaum, 2005.

[Swendsen and Wang, 1987] Robert H. Swendsen and Jian-Sheng Wang. Nonuniversal critical dynamics in Monte Carlo simulations. *Phys. Rev. Lett.*, 58(2):86–88, Jan 1987.

[Teh *et al.*, 2006] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *J Amer Stat Assoc*, 2006.

[Tu, 2005] Zhuowen Tu. An integrated framework for image segmentation and perceptual grouping. In *ICCV*, 2005.

[Wallach *et al.*, 2009] Hanna Wallach, David Mimno, and Andrew McCallum. Rethinking lda: Why priors matter. In *NIPS*, 2009.

[Yao *et al.*, 2009] L. Yao, D. Mimno, and A. McCallum. Efficient methods for topic model inference on streaming document collections. In *KDD*, 2009.