Shared Components Topic Models

Matthew R. Gormley **Mark Dredze** **Benjamin Van Durme** **Jason Eisner**
Center for Language and Speech Processing
Human Language Technology Center of Excellence
Department of Computer Science
Johns Hopkins University, Baltimore, MD
{mrg,mdredze,vandurme,jason}@cs.jhu.edu

Abstract

With a few exceptions, extensions to latent Dirichlet allocation (LDA) have focused on the distribution over topics for each document. Much less attention has been given to the underlying structure of the topics themselves. As a result, most topic models generate topics independently from a single underlying distribution and require millions of parameters, in the form of multinomial distributions over the vocabulary. In this paper, we introduce the Shared Components Topic Model (SCTM), in which each topic is a normalized product of a smaller number of underlying component distributions. Our model learns these component distributions and the structure of how to combine subsets of them into topics. The SCTM can represent topics in a much more compact representation than LDA and achieves better perplexity with fewer parameters.

1 Introduction

Topic models are probabilistic graphical models meant to capture the semantic associations underlying corpora. Since the introduction of latent Dirichlet allocation (LDA) (Blei et al., 2003), these models have been extended to account for more complex distributions over topics, such as adding supervision (Blei and McAuliffe, 2007), non-parametric priors (Blei et al., 2004; Teh et al., 2006), topic correlations (Li and McCallum, 2006; Mimno et al., 2007; Blei and Lafferty, 2006) and sparsity (Williamson et al., 2010; Eisenstein et al., 2011).

While much research has focused on modeling distributions over topics, less focus has been given to the makeup of the topics themselves. This emphasis leads us to find two problems with LDA and its variants mentioned above: (1) independently generated topics and (2) overparameterized models.

Independent Topics In the models above, the topics are modeled as independent draws from a single underlying distribution, typically a Dirichlet. This violates the topic modeling community's intuition that these distributions over words are often related. As an example, consider a corpus that supports two related topics, *baseball* and *hockey*. These topics likely overlap in their allocation of mass to high probability words (e.g. team, season, game, players), even though the two topics are unlikely to appear in the same documents. When topics are generated independently, the model does not provide a way to capture this sharing between related topics. Many extensions to LDA have addressed a related issue, LDA's inability to model topic correlation,[1] by changing the distributions over topics (Blei and Lafferty, 2006; Li and McCallum, 2006; Mimno et al., 2007; Paisley et al., 2011). Yet, none of these change the underlying structure of the topic's distributions over words.

Overparameterization Topics are most often parameterized as multinomial distributions over words: increasing the topics means learning new multinomials over large vocabularies, resulting in models consisting of millions of parameters. This issue was partially addressed in SAGE (Eisenstein et al., 2011) by encouraging sparsity in the topics which are parameterized by their difference in log-frequencies from a fixed background distribution. Yet the problem of overparameterization is also tied

[1] Two correlated topics, e.g. *nutrition* and *exercise*, are likely to co-occur, but their word distributions might not overlap.

to the number of topics, and though SAGE reduces the number of non-zero parameters, it still requires a vocabulary-sized parameter vector for each topic.

We present the Shared Components Topic Model (SCTM), which addresses both of these issues by generating each topic as a normalized product of a smaller number of underlying components. Rather than learning each new topic from scratch, we model a set of underlying component distributions that constrain topic formation. Each topic can then be viewed as a combination of these underlying components, where in a model such as LDA, we would say that components and topics stand in a one to one relationship. The key advantages of the SCTM are that it can learn and share structure between overlapping topics (e.g. *baseball* and *hockey*) and that it can represent the same number of topics in a much more compact representation, with far fewer parameters.

Because the topics are products of components, we present a new training algorithm for the significantly more complex product case which relies on a Contrastive Divergence (CD) objective. Since SCTM topics, which are products of distributions, could be represented directly by distributions as in LDA, our goal is not necessarily to learn better topics, but to learn models that are substantially smaller in size and generalize better to unseen data. Experiments on two corpora show that our model uses fewer underlying multinomials and still achieves lower perplexity than LDA, which suggests that these constraints could lead to better topics.

2 Shared Components Topic Models

The Shared Components Topic Model (SCTM) follows previous topic models in inducing admixture distributions of topics that are used to generate each document. However, here each topic multinomial distribution over words itself results from a normalized product of shared components, each a multinomial over words. Each topic selects a subset of components. We begin with a review and then introduce the SCTM.

Latent Dirichlet allocation (LDA) (Blei et al., 2003) is a probabilistic topic model which defines a generative process whereby sets of observations are generated from latent topic distributions. In the SCTM, we use the same generative process of topic assignments as LDA, but replace the K independently generated topics (multinomials over words) with products of C components.

Latent Dirichlet allocation generative process

For each topic $k \in \{1, \ldots, K\}$:
  $\boldsymbol{\phi}_k \sim \text{Dir}(\boldsymbol{\beta})$ [*draw distribution over words*]
For each document $m \in \{1, \ldots, M\}$:
  $\boldsymbol{\theta}_m \sim \text{Dir}(\boldsymbol{\alpha})$ [*draw distribution over topics*]
  For each word $n \in \{1, \ldots, N_m\}$:
    $z_{mn} \sim \text{Mult}(1, \boldsymbol{\theta}_m)$ [*draw topic*]
    $x_{mn} \sim \boldsymbol{\phi}_{z_{mi}}$ [*draw word*]

LDA draws each topic $\boldsymbol{\phi}_k$ independently from a Dirichlet. The model generates each document m of length M, by first sampling a distribution over topics $\boldsymbol{\theta}_m$. Then, for each word n, a topic z_{mn} is chosen and a word type x_{mn} is generated from that topic's distribution over words $\boldsymbol{\phi}_{z_{mi}}$.

A **Product of Experts** (PoE) model (Hinton, 1999) is the normalized product of the expert distributions. In the SCTM, each component (an expert) models an underlying multinomial word distribution. We let $\boldsymbol{\phi}_c$ be the parameters of the cth component, where ϕ_{cv} is the probability of the cth component generating word v. If the structure of a PoE included only components $c \in \mathcal{C}$ in the product, it would have the form: $p(x|\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_C) = \frac{\prod_{c \in \mathcal{C}} \phi_{cx}}{\sum_{v=1}^{V} \prod_{c \in \mathcal{C}} \phi_{cv}}$, where there are C components, and the summation in the denominator is over the vocabulary. In a PoE, each component can overrule the others by giving low probability to some word. A PoE can be viewed as a soft intersection of its components, whereas a mixture is a soft union.

The **Beta-Bernoulli model** (Griffiths and Ghahramani, 2006) is a distribution over binary matrices with a fixed number of rows and columns. It is the finite counterpart to the Indian Buffet Process. In this work, we use the Beta-Bernoulli as our prior for an unobserved binary matrix B with C columns and K rows. In the SCTM, each row $\boldsymbol{b}_k$ of the matrix, a binary feature vector, defines a topic distribution. The binary vector acts as a selector for the *structure* of the PoE for that topic. The row determines which components to include in the product by which entries b_{kc} are "on" (equal to 1) in that row. Under Beta-Bernoulli prior, for each column, a coin with weight π_c is chosen. For each entry in the column, the coin is flipped to determine if the entry is "on" or "off". This corresponds to

the notion that some components are *a priori* more likely to be included in topics.

The Beta-Bernoulli model generative process

For each component $c \in \{1, \ldots, C\}$: [*columns*]
  $\pi_c \sim \text{Beta}(\frac{\gamma}{C}, 1)$ [*draw probability of component c*]
  For each topic $k \in \{1, \ldots, K\}$: [*rows*]
    $b_{kc} \sim \text{Bernoulli}(\pi_c)$ *[draw whether topic includes cth component in its PoE]*

2.1 Shared Components Topic Models

The Shared Components Topic Model generates each document just like LDA, the only difference is the topics are not drawn independently from a Dirichlet prior. Instead, topics are soft intersections of underlying components, each of which is a multinomial distribution over words. These components are combined via a PoE model, and each topic is constructed according to a length C binary vector $\boldsymbol{b}_k$; where $b_{kc} = 1$ includes and $b_{kc} = 0$ excludes component c. Stacking the K vectors forms a $K \times C$ matrix; rows correspond to topics and columns to components. Overlapping topics share components in common.

Generative process SCTM's generative process generates topics and words, but must also generate the binary matrix. For each of the C shared components, we generate a distribution $\boldsymbol{\phi}_c$ over the V words from a Dirichlet parametrized by $\boldsymbol{\beta}$. Next, we generate a $K \times C$ binary matrix using the Beta-Bernoulli prior. These components and the binary matrix implicitly define the complete set of K topic distributions, each of which is a PoE.

$$p(x|\boldsymbol{b}_k, \boldsymbol{\phi}) = \frac{\prod_{c=1}^{C} \phi_{cx}^{b_{kc}}}{\sum_{v=1}^{V} \prod_{c=1}^{C} \phi_{cv}^{b_{kc}}} \qquad (1)$$

The distribution $p(\cdot|\boldsymbol{b}_k, \boldsymbol{\phi})$ defines the kth topic. Conditioned on these K topics, the remainder of the generative process, which generates the documents, is just like LDA.

The Shared Components Topic Model generative process

For each component $c \in \{1, \ldots, C\}$:
  $\boldsymbol{\phi}_c \sim \text{Dir}(\boldsymbol{\beta})$ [*draw distribution over words*]
  $\pi_c \sim \text{Beta}(\frac{\gamma}{C}, 1)$ [*draw probability of component c*]
  For each topic $k \in \{1, \ldots, K\}$:
    $b_{kc} \sim \text{Bernoulli}(\pi_c)$ *[draw whether topic includes cth component in its PoE]*
For each document $m \in \{1, \ldots, M\}$
  $\boldsymbol{\theta}_m \sim \text{Dir}(\boldsymbol{\alpha})$ [*draw distribution over topics*]
  For each word $n \in \{1, \ldots, N_m\}$
    $z_{mn} \sim \text{Mult}(1, \boldsymbol{\theta}_m)$ [*draw topic*]
    $x_{mn} \sim p(\cdot|\boldsymbol{b}_{z_{mn}}, \boldsymbol{\phi})$ given by Eq. (1) [*draw word*]

See Figure 1 for the graphical model.

Discussion An advantage of this formulation is the ability to model many topics using few components. While LDA must maintain $V \times K$ parameters for the topic distributions, the SCTM maintains just $V \times C$ parameters, plus an additional $K \times C$ binary matrix. Since $C < K \ll V$ this results in many fewer parameters for the SCTM.[2] Extending the number of topics (rows) requires storing additional binary vectors, a lightweight requirement. In theory, we could enable all 2^C possible component combinations, although we expect to use far less. On the other hand, constraining the SCTM's topics by the components gives less flexible topics as compared to LDA. However, we find empirically that a large number of topics can be effectively modeled with a smaller number of components.

Observe that we can reparameterize the SCTM as LDA by assuming an identity square matrix; each component corresponds to a topic in LDA, making LDA a special case of the SCTM with an identity matrix I_C. Intuitively, SCTM learning could produce an LDA model where appropriate. Finally, we can also think of the SCTM as learning the structure of many PoE models. In applications where experts abstain, the SCTM could learn in which setting (row) each expert casts a vote.

3 Parameter Estimation

Parameter estimation infers values for model parameters $\boldsymbol{\phi}$, $\boldsymbol{\pi}$, and $\boldsymbol{\theta}$ from data using an unsupervised training procedure. Because exact inference is intractable in the SCTM, we turn to approximate methods. As is common in these models, we will integrate out $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$, sample latent variables Z and B, and optimize the components $\boldsymbol{\phi}$. Our algorithm follows the outline of the Monte Carlo EM (MCEM) algorithm (Wei and Tanner, 1990). In the Monte Carlo E-step, we will re-sample the latent variables Z and B based on current model parameters $\boldsymbol{\phi}$ and observed data X. In the M-step, we will find new model parameters $\boldsymbol{\phi}$. Since these parameters correspond to experts in the PoE, we rely on a contrastive divergence (CD) objective (Hinton, 2002), popular for PoE training, rather than maximizing the data

[2] The vocabulary size V could be much larger if n-grams or relational triples are used, as opposed to unigrams.

log-likelihood. Normally, CD only estimates the parameters of the expert distributions. However, in our model, the structure of the PoEs themselves change based on the E-step. Since we generate multiple samples in the E-step, we modify the CD objective to compute the gradient for each E-step sample and take the average to approximate the expectation under B and Z.[3]

3.1 E-Step

The E-step approximates an expectation under $p(B, Z|X, \boldsymbol{\phi}, \boldsymbol{\alpha}, \gamma)$ for latent topic assignments Z and matrix B using Gibbs sampling. The Gibbs sampler uses the full conditionals for both z_i (7) and b_{kc} (12), which we derive in Appendix A. Using this sampler, we obtain J samples of Z and B by iterating through each value of z_i and b_{kc} J times (in our experiments, we use J=1, which appears to work as well on this task as multiple samples). These J samples are then used in the M-step as an approximation of the expectation of the latent variables.

3.2 M-Step

Given many samples of B and Z, the M-step optimizes the component parameters $\boldsymbol{\phi}$ which cannot be collapsed out. We utilize the standard PoE training procedure for experts: contrastive divergence (CD). We approximate the CD gradient as the difference of the data distribution and the one-step reconstruction of the data according to the current parameters. As in Generalized EM (Dempster et al., 1977), a single gradient step in the direction of the contrastive divergence objective is sufficient for each M-step. A key difference in our model is that we must incorporate the expectation of the PoE model structure, which in our case is a random variable instead of a fixed observed structure. We achieve this by simply computing the CD gradient for each PoE given each of the J samples $\{Z, B\}^{(j)}$ from the E-Step, then average the result.

Another difficulty arises from computing the gradient directly for the multinomial $\boldsymbol{\phi}_c$ due to the $V-1$ degrees of freedom imposed by sum-to-one constraints. Therefore, we switch to the *natural parameters*, which obviates the need for considering the sum-to-one constraint in the optimization, by defining $\boldsymbol{\phi}_c$ in terms of V real valued parameters $\{\xi_{c1}, \ldots, \xi_{cV}\}$:

$$\phi_{cv} = \frac{\exp(\xi_{cv})}{\sum_{t=1}^{V} \exp(\xi_{cv})} \tag{2}$$

The V parameters ξ_{cv} are then used to compute ϕ_{cv} for use in the E-step.

As explained above, the M-step does not maximize the data log-likelihood, but instead minimizes contrastive divergence. Hinton (2002) explains that maximizing data log-likelihood is equivalent to minimizing $Q^0||Q_\xi^\infty$, the KL divergence between the observed data distribution, Q^0, and the model's equilibrium distribution, Q_ξ^∞.[4] Minimizing $Q^0||Q_\xi^\infty$ would require the computation of an intractable expectation under the equilibrium distribution. We avoid this by instead minimizing the contrastive divergence objective,

$$\text{CD}(\xi|\{Z, B\}^{(j)}) = Q^0||Q_\xi^\infty - Q_\xi^1||Q_\xi^\infty, \tag{3}$$

where Q_ξ^1 is the distribution over *one-step* reconstructions of the data, X given Z, B, ξ, that are generated by a single step of Gibbs sampling.

Unlike standard applications of CD training, the hidden variables (Z,B) are not contained within the experts. Instead they define the *structure* of the PoE model, where B indicates which experts to use in each product (topic) and Z indicates which PoE generates each word. Unfortunately, CD training cannot infer this structure since the CD derivation makes use of a fixed structure in the one-step reconstruction. Therefore, we have taken a MCEM approach, first sampling the PoE structure in the E-step, then

[3] CD training within MCEM is not the only possible approach. One alternative would be to compute the CD gradient summing over all values of B and Z, effectively training the entire model using CD. This approach prevents the normal CD objective derivation from being simplified into a more tractable form. Another approach would be a pure MCMC algorithm, which sampled $\boldsymbol{\phi}$ directly. While using the natural parameters allows the sampler to mix, it is too computationally intensive to be practical. Finally, we could train with Generalized MCEM, where the exact gradient of the log-likelihood (or log-posterior) is used, but this easily gets stuck in local minima. After experimenting with these and other options, we present our current most effective estimation method.

[4] Hinton (2002) used this notation because the data distribution, Q^0, can be described as the state of a Markov chain at time 0 that was started at the data distribution. Similarly, the equilibrium distribution, Q_ξ^∞ could be obtained by running the same Markov chain to time ∞.

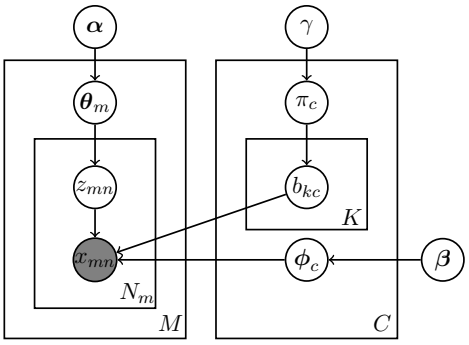


Figure 1: The graphical model for the SCTM.

fixing these samples for Z and B when computing the *one-step* reconstruction of the data, X.

Contrastive Divergence Gradient We provide the approximate derivative of the contrastive divergence objective, where Z and B are treated as fixed.[5]

$$\frac{d\,\mathrm{CD}(\xi|\{Z,B\}^{(j)})}{d\xi} \approx -\left\langle \frac{d\log f(x|\boldsymbol{b}_z,\phi)}{d\xi} \right\rangle_{Q^0} + \left\langle \frac{d\log f(x|\boldsymbol{b}_z,\phi)}{d\xi} \right\rangle_{Q^1_\xi}$$

where $f(x|\boldsymbol{b}_z,\phi) = \prod_{c=1}^{C} \phi_{cx}^{b_{zc}}$ is the numerator of $p(x|\boldsymbol{b}_z,\phi)$ and the derivative of its log is efficient to compute:

$$\frac{d\,\log f(x|\boldsymbol{b}_z,\phi)}{d\xi_{cv}} = \begin{cases} b_{zc}(1-\phi_{cv}) & \text{for } x = v \\ -b_{zc}\phi_{cv} & \text{for } x \neq v \end{cases}$$

To approximate the expectation under Q^1_ξ, we hold Z, B, ξ fixed and resample the data, X, using one step of Gibbs sampling.

3.3 Summary

Our learning algorithm can be viewed in terms of a Q function: $Q(\xi|\xi^{(t)}) \approx \frac{1}{J}\sum_{j=1}^{J} \mathrm{CD}(\xi|\{Z,B\}^{(j)})$where we average over J samples. The E-step computes $Q(\xi|\xi^{(t)})$. The M-step minimizes Q with respect to ξ to obtain the updated $\xi^{(t+1)}$ by performing gradient descent on the Q function as $\xi_{cv}^{(t+1)} = \xi_{cv}^{(t)} - \eta \cdot \frac{d\,Q(\xi|\xi^{(t)})}{d\xi_{cv}}$ for all values of c, v.

[5]The derivative is approximate because we drop the term: $-\frac{d\,Q^1_\xi}{d\xi} \cdot \frac{d\,Q^1_\xi || Q^\infty_\xi}{d\,Q^1_\xi}$, which is 'problematic to compute' (Hinton, 2002). This is the standard use of CD.

Algorithm 1 SCTM Training

Initialize parameters: ξ_c, b_{kc}, z_i.
while not converged **do**
  {E-step:}
  for $j = 1$ to J **do**
    {Draw jth sample $\{Z,B\}^{(j)}$}
    for $i = 1$ to N **do**
      Sample z_i using Eq. (7)
    for $k = 1$ to K **do**
      for $c = 1$ to C **do**
        Sample b_{kc} using ratio in Eq. (12)
  {M-step:}
  for $c = 1$ to C **do**
    for $v = 1$ to V **do**
      Single gradient step over ξ

$$\xi_{cv}^{(t+1)} = \xi_{cv}^{(t)} - \eta \cdot \frac{d\,Q(\boldsymbol{\phi}|\boldsymbol{\phi}^{(t)})}{d\xi_{cv}}$$

4 Related Models

The SCTM is closely related to the the Infinite Overlapping Mixture Model (IOMM) (Heller and Ghahramani, 2007), yet our model differs from and, in some ways, extends theirs. The IOMM models the geometric overlap of Gaussian clusters using PoEs, and models the structure of the PoEs with the rows of a binary matrix. The SCTM models a finite number of columns, where the IOMM models an infinite number. The IOMM generates a row for each data point, whereas the SCTM generates a row for each topic. Thus, the SCTM goes beyond the IOMM by allowing the rows to be shared among documents and models document-specific mixtures over the rows of the matrix.[6]

SAGE for topic modeling (Eisenstein et al., 2011) can be viewed as a restricted form of the SCTM. Consider an SCTM in which the binary matrix is restricted such that the first column, $b_{\cdot,1}$, consists of all ones and the remainder forms a diagonal matrix. If we then set the first component, $\boldsymbol{\phi}_1$, to the corpus background distribution, and add a Laplace prior on the natural parameters, ξ_{cv}, we have the SAGE model. Note that by removing the restriction that the matrix contain a diagonal, we could allow multiple components to combine in the SCTM fashion, while incorporating SAGE's sparsity benefits.

[6]The IOMM uses Metropolis-Hastings (MH) to sample the parameters of the experts. This approach is computationally feasible because their experts are Gaussian, unlike the SCTM in which the experts are multinomials and the MH step too expensive.

The relation of TagLDA (Zhu et al., 2006) to the SCTM is similar to that of SAGE and SCTM. TagLDA has a PoE of exactly two experts: one expert for the topic, and one for the supervised word-level tag. Examples of tags are *abstract* or *body*, indicating which part of a research paper the word appears in.

Unlike the SCTM and SAGE, most prior extensions to LDA have enhanced the distribution over topics for each document. One of the closest is *hierarchical LDA* (hLDA) (Blei et al., 2004) and its application to PAM (Mimno et al., 2007). Though topics are still generated independently from a Dirichlet prior, hLDA learns a tree structure underlying the topics. Each document samples a single path through the tree and samples words from topics along that path. The SCTM models an orthogonal issue to topic hierarchy: how the topics themselves are represented as the intersection of components. Finally, while prior work has primarily used mixtures for the sake of conjugacy, we take a fundamentally different approach to modeling the structure by using normalized product distributions.

5 Evaluation

We compare the SCTM with LDA in terms of overall model performance (held-out perplexity) as well as parameter usage (varying numbers of components and topics). We select LDA as our baseline since our model differs only in how it forms topics, which focuses evaluation on the benefit of this model change.

We consider two popular data sets for comparison: NIPS: A collection of 1,617 NIPS abstracts from 1987 to 1999[7], with 77,952 tokens and 1,632 types. 20NEWS: 1,000 randomly selected articles from the 20 Newsgroups dataset,[8] with 70,011 tokens and 1,722 types. Both data sets excluded stop words and words occurring in fewer than 10 documents. For 20NEWS, we used the standard by-date train/test split. For NIPS, we randomly partitioned the data by document into 75% train and 25% test.

We compare the SCTM to LDA by evaluating the average perplexity-per-word of the held-out test data, $\text{perplexity} = 2^{-\log_2(\text{data}|\text{model})/N}$. Exact computation is intractable, so we use the *left-to-right algorithm* (Wallach et al., 2009) as an accurate alternative. With the topics fixed, the SCTM is equivalent to LDA and requires no adaptation of the left-to-right algorithm.

We used a collapsed Gibbs sampler for training LDA and the algorithm described above for training the SCTM. Both were trained for 4000 iterations, sampling topics every 10 iterations after a burn-in of 3000. The hyperparameter $\boldsymbol{\alpha}$ was optimized as an asymmetric Dirichlet, $\boldsymbol{\beta}$ as a symmetric Dirichlet, and $\gamma = 3.0$ was fixed.[9] Following the observation of Hinton (2002) that CD training benefits from initializing the experts to nearly uniform distributions, we initialize the component distributions from a symmetric Dirichlet with parameter $\hat{\boldsymbol{\beta}} = 1 \times 10^6$. We use $J = 1$ samples per iteration and a decaying learning rate centered at $\eta = 100$.[10] We ranged LDA from 10 to 200 topics, and the SCTM from 10 to 100 components (C). We then selected the number of SCTM topics (K) as $K \in \{C, 2C, 3C, 4C, 5C\}$. For each model, we used five random restarts, selecting the model with the highest training data likelihood.

5.1 Results

Our goal is to demonstrate that (1) modeling topics as products of components is an expressive alternative to generating topics independently and (2) the SCTM can both achieve lower perplexity than LDA and use fewer model parameters in doing so.

Topics as Products of Components Figures 3b and 3c show the perplexity for the held-out portions of 20NEWS and NIPS for different numbers of components C. The shaded region shows the full SCTM perplexity range we observed for different K and at each value of C, we label the number of topics K (rows in the binary matrix). For each number of components, LDA falls within the upper portion of the shaded region. While for some (small) values of K for the SCTM, LDA does better, the SCTM can easily include more K (requiring few new parameters) to achieve better results. This supports our hypothesis that topics can be comprised of the overlap between shared underlying components. More-

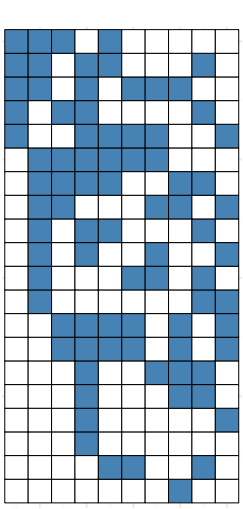
[7] We follow prior work (Blei et al., 2004; Li and McCallum, 2006; Li et al., 2007) in using only the abstracts: http://www.cs.nyu.edu/~roweis/data.html

[8] Williamson et al. (2010) created a similar subset: http://people.csail.mit.edu/jrennie/20Newsgroups/

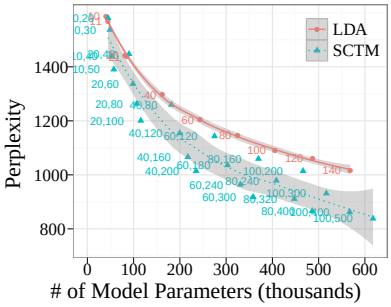
[9] On development data the model was rather insensitive to γ.

[10] We experimented with larger J but it had no effect.

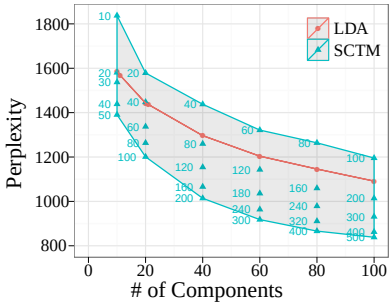
Figure 2: SCTM binary matrix and topics from 3599 training documents of 20NEWS for $C = 10$, $K = 20$. Blue squares are "on" (equal to 1).



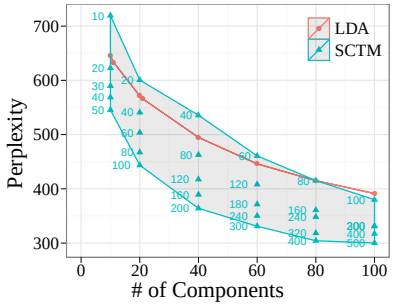
k	α_k	Top words for topic	Top words for topic after ablating component c=1
1	0.306	subject organization israel return define law org	organization subject israel law peace define israeli
2	0.031	encryption chip clipper keys des escrow security law	administration president year market money senior
3	0.025	turkish armenian armenians war turkey turks armenia	years food center year air russian war army
4	0.102	drive card disk scsi hard controller mac drives	opinions drive hard power support cost research price
5	0.071	image jpeg window display code gif color mit	pitt file program year center programs image division
6	0.018	jews israeli jewish arab peace land war arabs	
7	0.074	org money back question years thing things point	
8	0.106	christian bible church question christ christians life	
9	0.011	administration president year market money senior	
10	0.055	health medical center research information april	
11	0.063	gun law state guns control bill rights states	
12	0.160	world organization system israel state usa cwru reply	
13	0.042	space nasa gov launch power wire ground air	
14	0.038	space nasa gov launch power wire ground air	
15	0.079	team game year play games season players hockey	
16	0.158	car lines dod bike good uiuc sun cars	
17	0.136	windows file government key jesus system program	
18	0.122	article writes center page harvard virginia research	
19	0.017	max output access digex int entry col line	
20	0.380	lines people don university posting host nntp time	



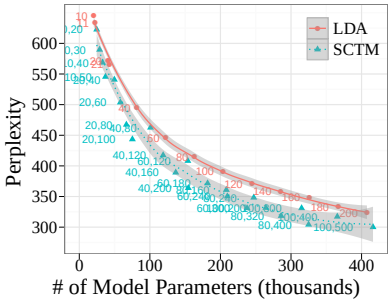
(a)



(b)



(c)



(d)

Figure 3: Perplexity results on held-out data for 20NEWS (b) and NIPS (c) showing the results of LDA and the SCTM for the same number of components and varying K (SCTM). For the same number of components (multinomials), the SCTM achieves lower perplexity by combining them into more topics. Results for 20NEWS (a) and NIPS (d) showing non-square SCTM achieves lower perplexity than LDA with a more compact model.

over, this suggests that our products (PoEs) provide additional and complementary expressivity over just mixtures of topics.

Model Compactness Including an additional topic in the SCTM only adds C binary parameters, for an extra row in the matrix. Whereas in LDA, an additional topic requires V (the size of the vocabulary) additional parameters to represent the multinomial. In both cases, the number of document-specific parameters must increase as well. Figures 3a and 3d present held-out perplexity vs. number of model parameters on 20NEWS and NIPS, excluding the case of square ($C = K$) binary matrices for the SCTM. The regions show a confidence interval ($p = 0.05$) around the smoothed fit to the data, LDA labels show C, and SCTM labels show C, K. The SCTM achieves lower perplexity with fewer model parameters, even when the increase in non-component parameters is taken into account. We expect that because of its smaller size the SCTM exhibits lower sample complexity, allowing for better generalization to unseen data.

5.2 Analysis

Figure 2 gives the binary matrix and topics learned on a larger section of 20NEWS training documents. These topics evidence that the SCTM is able to achieve a diversity of topics by combining various subsets of components, and we expect that the low perplexity achieved by the SCTM can be attributed

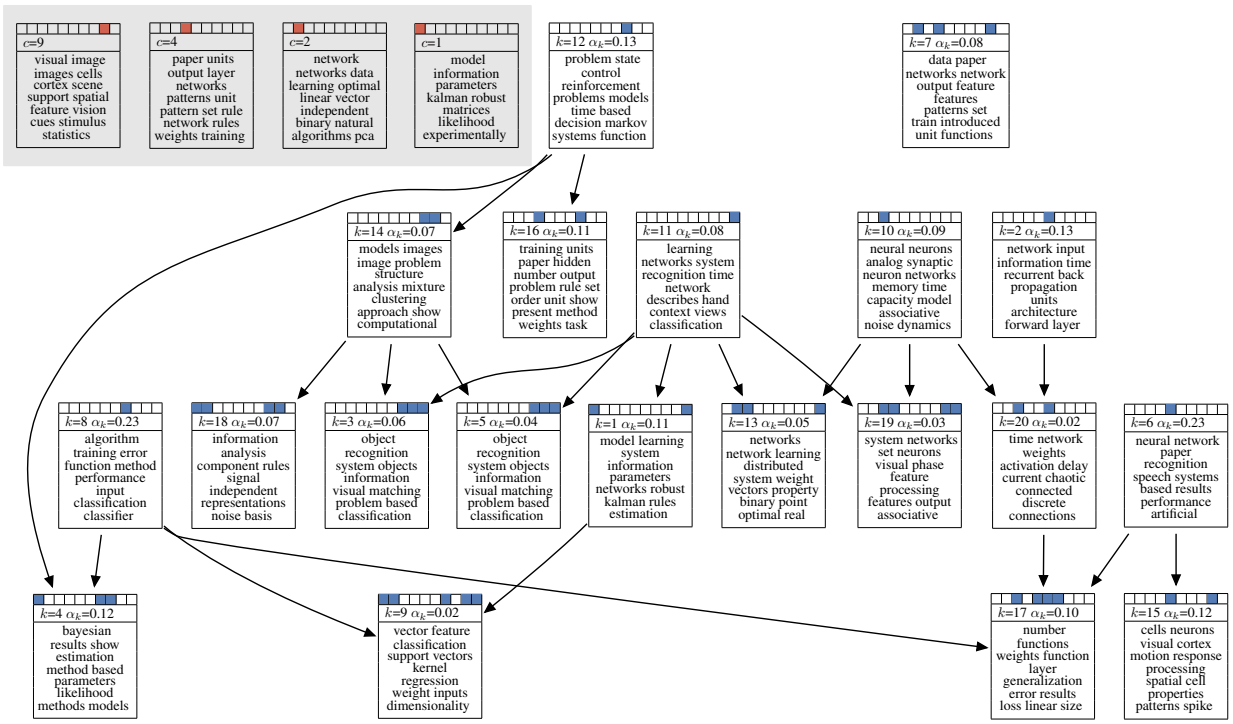


Figure 4: Hasse diagram on NIPS for $C = 10$, $K = 20$ showing the top words for topics and unrepresented components (in shaded box). Notice that some topics only consist of a single component. The shaded box contains the components that didn't appear as a topic. For the sake of clarity, we only show arrows for the subsumption relationships between the topics, and we omit the implicit arrows between the components in the shaded box and the topics.

to the high-level of component re-use across topics.

Topics are typically interpreted by looking at the top-N words, whereas the top-N words of a component often do not even appear in the topics to which it contributes. Instead, we find that the components contribution to a topic is typically through vetoing words. For example, the top words of component c=1, corresponding to the first column of the binary matrix in figure 2, are [subject organization posting apple mit screen write window video port], yet only a few of these appear in topics k=1,2,3,4,5, which use it.

On the right of figure 2, we show what the topics become when we ablate component c=1 from the matrix by setting the column to all zeros. Topic k=2 changes from being about *information security* to *general politics* and is identical to k=9. Topic k=3 changes from *the Turkish-Armenian War* to a more general *war* topic. Topic k=4 changes to a less focused version of itself. In this way, we can gain further insight into the contribution of this component, and the way in which components tend to increase the specificity of a topic to which they are added.

The SCTM learns each topic as a soft intersection of its components, as represented by the binary matrix. We can describe the overlap between topics based on the components that they have in common. One topic subsumes another topic when the parent consists of a subset of the child's components. In this way, the binary matrix defines a Hasse diagram, a directed acyclic graph describing all the subsumption relationships between topics. Figure 4 shows such a Hasse diagram on the NIPS data. Several topics consist of only a single component, such as k=12 on *reinforcement learning* and k=8 on *optimization*. These two topics combine with the component c=1 so that their overlap forms the topic k=4 on *Bayesian methods*. These subsumption relationships are different from and complementary to hLDA (see §4), which models topic co-occurrence, not component intersection. For example, topic k=10 on *connectionism* and k=2 on *neural networks* intersect to form k=20 which contains words that would only appear in *both* of its subsuming topics, thereby explicitly modeling topic overlap.

The SCTM sometimes learns identical topics (two rows with the same binary entries "on") such as k=13 and k=14 in figure 2 and k=3 and k=5 in figure 4, which is likely due to the Gibbs sampler for the binary matrix getting stuck in a local optimum.

6 Discussion

We have presented the Shared Components Topic Model (SCTM), in which topics are products of underlying component distributions. This model change learns shared topic structures—as expressed through components—as opposed to generating each topic independently. Reducing the number of components yields more compact models with lower perplexity than LDA. The two main limitations of the current SCTM are, when restricted to a square binary matrix ($C = K$), the inference procedure is unable to recover a model with perplexity as low as a collapsed Gibbs sampler for LDA, and the components are not consistently interpretable.

The use of components opens up interesting directions of research. For example, task specific side information can be expressed as priors or constraints over the components, or by adding conditioning variables tied to the components. Additionally, tasks beyond document modeling may benefit from representing topics as products of distributions. For example, in vision, where topics are classes of objects, the components could be features of those objects. For selectional preference, components could correspond to semantic features that intersect to define semantic classes (Gormley et al., 2011). We hope new opportunities will arise as this work explores a new research area for topic models.

Appendix A: Derivation of Full Conditionals

The model's complete data likelihood over all variables—observed words X, latent topic assignments Z, matrix B, and component/expert distributions $\boldsymbol{\phi}$:

$$p(X, Z, B, \boldsymbol{\phi}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma) = p(X|Z, B, \boldsymbol{\phi})p(Z|\boldsymbol{\alpha})p(B|\gamma)p(\boldsymbol{\phi}|\boldsymbol{\beta}) \quad (4)$$

This follows from the conditional independence assumptions. It is tractable to integrate out all parameters except $Z, B, \boldsymbol{\phi}$ and hyperparameters $\boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma$. [11]

Full conditional of z_i Recall that $p(Z|\boldsymbol{\alpha})$ is the Dirichlet-Multinomial distribution over topic assignments, where $\boldsymbol{\theta}$ has been integrated out. The form of this distribution is identical to the corresponding distribution over topics in LDA. The derivation of the full conditional of $z_i \in \{1, \ldots, K\}$, follows from the factorization in Eq. 4:

$$p(z_i|X, Z^{-(i)}, B, \boldsymbol{\phi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma) \quad (5)$$
$$\propto p(X|Z, B, \boldsymbol{\phi})p(Z|\boldsymbol{\alpha}) \quad (6)$$
$$\propto p(x_i|\boldsymbol{b}_{z_i}, \boldsymbol{\phi})(\tilde{n}^{-(i)}_{mz_i} + \alpha_{z_i}) \quad (7)$$

$Z^{-(i)}$ is the set of all topic assignments except z_i. We use the independence of each document, recalling that example i belongs to document m. In practice, we cache $p(x|\boldsymbol{b}_z, \boldsymbol{\phi})$ for all x, z ($V \times K$ values) and these are shared by all z_i in a sampling iteration.

Above, just as in LDA, $p(Z|\boldsymbol{\alpha})$ is simplified by proportionality to $(\tilde{n}^{-(i)}_{mz_i} + \alpha_{z_i})$, where $\tilde{n}^{-(i)}_{mk}$ is the count of examples for document m that are assigned topic k excluding z_i's contribution (Heinrich, 2008).

Full conditional of b_{kc} Recall that $p(B|\gamma)$ is the prior for a Beta-Bernoulli matrix. The full conditional distribution of a position in the binary vector is (Griffiths and Ghahramani, 2006):

$$p(b_{kc} = 1|B^{-(kc)}, \gamma) = \frac{\bar{n}^{-(k)}_c + \frac{\gamma}{C}}{K + \frac{\gamma}{C}} \quad (8)$$

where $\bar{n}^{-(k)}_c$ is the count of topics with component c excluding topic k, and $B^{-(kc)}$ is the entire matrix except for the entry b_{kc}.

To find the full conditional for $b_{kc} \in \{0, 1\}$, we again start with the factorization from Eq. 4.

$$p(b_{kc}|X, Z, B^{-(kc)}, \boldsymbol{\phi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma) \quad (9)$$
$$\propto p(X|Z, B, \boldsymbol{\phi})p(B|\gamma) \quad (10)$$
$$\propto \left[\prod_{i:z_i=k} p(x_i|\boldsymbol{b}_{z_i}, \boldsymbol{\phi})\right] p(b_{kc}|B^{-(kc)}, \gamma) \quad (11)$$

where $p(b_{kc}|B^{-(kc)}, \gamma)$ is given by Eq. 8,

$$= \left[\frac{\left(\prod_{v=1}^{V} \phi_{cv}^{\hat{n}_{kv}}\right)^{b_{kc}}}{\left(\sum_{v=1}^{V}\prod_{j=1}^{C} \phi_{jv}^{b_{kj}}\right)^{-||\hat{\boldsymbol{n}}_k||_1}}\right] p(b_{kc}|B^{-(kc)}, \gamma) \quad (12)$$

and where $\hat{n}_{kv}$ is the count of words assigned topic k that are type v, and $||\hat{\boldsymbol{n}}_k||_1$ (the L_1-norm of count vector $\hat{\boldsymbol{n}}_k$) is the count of *all* words with topic k.

[11] For simplicity, we switch from indexing examples as x_{mn} to x_i. In this presentation, x_i is the ith example in the corpus, which corresponds to some m, n pair.

References

David Blei and John Lafferty. 2006. Correlated topic models. In *Advances in Neural Information Processing Systems (NIPS)*, volume 18.

David Blei and Jon McAuliffe. 2007. Supervised topic models. In *Advances in Neural Information Processing Systems (NIPS)*.

David Blei, Andrew Ng, and Michael Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3.

David Blei, Thomas Griffiths, Michael Jordan, and Joshua Tenenbaum. 2004. Hierarchical topic models and the nested chinese restaurant process. In *Advances in Neural Information Processing Systems (NIPS)*, volume 16.

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

Jacob Eisenstein, Amr Ahmed, and Eric P. Xing. 2011. Sparse additive generative models of text. In *International Conference on Machine Learning (ICML)*.

Matthew R. Gormley, Mark Dredze, Benjamin Van Durme, and Jason Eisner. 2011. Shared components topic models with application to selectional preference. In *Learning Semantics Workshop at NIPS 2011*, December.

Thomas Griffiths and Zoubin Ghahramani. 2006. Infinite latent feature models and the indian buffet process. In *Advances in Neural Information Processing Systems (NIPS)*, volume 18.

Gregor Heinrich. 2008. Parameter estimation for text analysis. Technical report, Fraunhofer IGD.

Katherine A. Heller and Zoubin Ghahramani. 2007. A nonparametric bayesian approach to modeling overlapping clusters. In *Artificial Intelligence and Statistics (AISTATS)*, pages 187–194.

Geoffrey Hinton. 1999. Products of experts. In *International Conference on Artificial Neural Networks (ICANN)*.

Geoffrey Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800.

Wei Li and Andrew McCallum. 2006. Pachinko allocation: DAG-structured mixture models of topic correlations. In *International Conference on Machine Learning (ICML)*, pages 577–584.

Wei Li, David Blei, and Andrew McCallum. 2007. Nonparametric bayes pachinko allocation. In *Uncertainty in Artificial Intelligence (UAI)*.

David Mimno, Wei Li, and Andrew McCallum. 2007. Mixtures of hierarchical topics with pachinko allocation. In *International Conference on Machine Learning (ICML)*, pages 633–640.

John Paisley, Chong Wang, and David Blei. 2011. The discrete infinite logistic normal distribution for Mixed-Membership modeling. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Yee Whye Teh, Michael Jordan, Matthew Beal, and David Blei. 2006. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581.

Hanna Wallach, Ian Murray, Ruslan Salakhutdinov, and David Mimno. 2009. Evaluation methods for topic models. In *International Conference on Machine Learning (ICML)*, pages 1105–1112.

Greg Wei and Martin Tanner. 1990. A monte carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85(411):699–704.

Sinead Williamson, Chong Wang, Katherine Heller, and David Blei. 2010. The IBP compound dirichlet process and its application to focused topic modeling. In *International Conference on Machine Learning (ICML)*.

Xiaojin Zhu, David Blei, and John Lafferty. 2006. TagLDA: bringing document structure knowledge into topic models. Technical Report TR-1553, University of Wisconsin.