Replicating First Analysis of Latent Dirichlet Allocation

Benjamin Xie
Massachusetts Institute of Technology
bxie@mit.edu

Jason Leung
Massachusetts Institute of Technology
jasonleu@mit.edu

Keywords

Generative Probabilistic Models, Topic Modeling, Text Analysis, Classification, Collaborative Filtering

1. INTRODUCTION

We analyze and replicate and extend experiments from the original paper on Latent Dirichlet Allocation (LDA), as presented by Blei, Ng, and Jordan [5]. This paper introduces LDA, which has since become a fundamental model for analyzing discrete data such as text to perform such actions including topic modeling. In this paper, we adopt the same topic modeling terminology:

- A *word* is the basic unit of discrete data, drawn from a vocabulary of words, { 1,...,V }.
- A *document* is a sequence of N words denoted by $\mathbf{w} = (w_1, w_2, ...w_N)$.
- A *corpus* is a collection of M documents denoted by $\mathbf{D} = \{\mathbf{w}_1, \mathbf{w}_2, ...\mathbf{w}_M\}$.

This allows us to understand the concept of a *topic*, a latent probability distribution over a collection of words.

2. LATENT DIRICHLET ALLOCATION

LDA is commonly used for topic modeling. Using a three-layer Bayesian model, LDA provides a probabilistic generative model that describes how documents in a dataset are created. LDA assumes that words w_i in each document, $\mathbf{w}$, in a corpus, $\mathbf{D}$, are generated in this process:

1. Choose $N \sim$ Poisson(ξ).
2. Choose $\theta \sim$ Dir(α).
3. For each of N words w_i
   (a) Choose a topic $z_n \sim$ Multinomial(θ).
   (b) Choose a word w_i from $p(w_i|z_n, \beta)$, i.e. $w_i \sim$ Multinomial(β_{z_n}).

In the basic generative model, several key assumptions are made. First, the total number of topics, k, which is the dimensionality of both the Dirichlet distribution and the topic variable z, are assumed known and fixed. Second, the length of vocabulary of words,V, is also assumed known, such that the matrix β, which parameterizes the word probabilities, w_i, has dimensions $k \times V$. Under this generative model, documents can contain multiples topics as z_n is sampled for each word within a document.

Also, since the order of words is ignored, there is an inherent *bag-of-words* assumption. This assumption of *exchangeability* for the words in the documents allows us to use De Finetti's representation theorem, which states that the joint distribution of an infinitely exchangeable sequence of random variables is as if a random parameter were drawn from some distribution and then the random variables in question were independent and identically distributed, conditioned on that parameter [5].

Infinite Exchangeability:

$$p(z_1, \ ... \ , z_N) = p(z_{\pi(1)}, \ ... \ , z_{\pi(N)})$$

where π is a permutation of integers from 1 to N.

Hence, given α and β, the joint distribution of $\mathbf{z}$ and $\mathbf{w}$ over N words in a document is:

$$p(\mathbf{z}, \mathbf{w}|\alpha, \beta) = \int p(\theta|\alpha) \Big(\prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n, \beta) \Big) d\theta$$

, such that the probability of an entire corpus simply sums over z and takes the product over all documents:

$$p(D|\alpha, \beta) = \prod_{d=1}^{M} \int p(\theta_d|\alpha) \Big(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_n|\theta_d) p(w_{dn}|z_{dn}, \beta) \Big) d\theta_d$$

Figure 1 shows a graphical representation of this generative model, where plates represent replicates of N words and M documents.

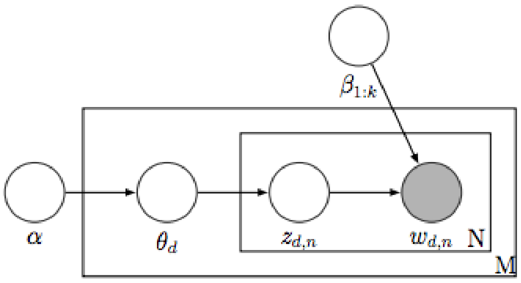


Figure 1: Graphical model representation of LDA's generative process

2.1 LDA Inference and Parameter Estimation

While the generative model assumes we know the K topic distributions both in α and β, the goal of topic modelling is to find these topic distributions given some documents. We then wish to find a probabilistic model of a corpus that assigns high probability to members of the corpus, but generalises well so that it also assigns high probability to other similar documents.

Thus, the goal is to determine the posterior distribution of the latent variables given the documents in the corpus. For each document, we have:

$$p(\theta, \mathbf{z}|\mathbf{w}, \alpha, \beta) = \frac{p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta)}{p(\mathbf{w}|\alpha, \beta)} \tag{1}$$

where,

$$\begin{aligned} p(\theta, \mathbf{z}|\mathbf{w},&\alpha, \beta) \\ &= p(\mathbf{w}|\mathbf{z}, \beta)p(\mathbf{z}|\theta)p(\theta|\alpha) \\ &= \Big(\prod_{n=1}^{N} \beta_{z_n,w_n}\Big)\Big(\prod_{n=1}^{N} \theta_{z_n}\Big)\Big(\frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \int (\prod_{i=1}^{k} \theta_i^{\alpha_i - 1})\Big) \\ &= \Big(\prod_{n=1}^{N} \beta_{z_n,w_n}\theta_{z_n}\Big)\Big(\frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \int (\prod_{i=1}^{k} \theta_i^{\alpha_i - 1})\Big) \\ &= \Big(\prod_{n=1}^{N}\prod_{i=1}^{k}\prod_{j=1}^{V} (\beta_{i,j}\theta_i)^{w_n^j z_n^j}\Big)\Big(\frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \prod_{i=1}^{k} \theta_i^{\alpha_i - 1}\Big) \end{aligned}$$

,and we marginalise over θ and $\mathbf{z}$ to get denominator:

$$p(\mathbf{z}|\alpha, \beta) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \int (\prod_{i=1}^{k} \theta_i^{\alpha_i - 1})\Big(\prod_{n=1}^{N}\sum_{i=1}^{k}\prod_{j=1}^{V} (\beta_{i,j}\theta_i)^{w_n^j}\Big) d\theta$$

But this makes equation 1 intractable because taking the log of the likelihood function doesn't separate the θ and β. Hence, we use variational inference.

2.1.1 Variational Inference

We use variational inference to find variational parameters to find the tightest possible lower bound on the log likelihood. Since we were faced with the coupling of θ with β in the previous section, we modify the original graphical model and drop the problematic edges (Figure 2).

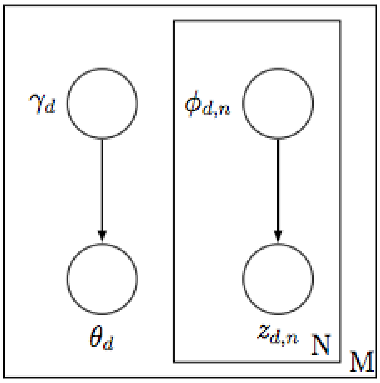


Figure 2: Graphical model representation of LDA's generative process

The variational distribution becomes:

$$q(\theta, \mathbf{z}|\gamma, \phi) = q(\theta|\gamma) \prod_{n=1}^{N} q(z_n|\phi_n)$$

,where the Dirichlet parameter γ and the multinomial parameters $(\phi_1, ...\phi_n)$ are the free variational parameters.

As shown in A.3 of Blei et al. [5], finding the lower bound on the log likelihood translates into the following optimization problem:

$$(\gamma^*, \phi^*) = \text{argmin}_{\gamma,\phi} D(q(\theta, \mathbf{z}|\gamma, \phi)||p(\theta, \mathbf{z}|\mathbf{w}, \alpha, \beta)$$

Hence, the values of the variational parameters can be found by minimizing the Kullback-Leibler (KL) divergence between the variational distribution and the actual posterior distribution. We thus get the following update equations:

$$\phi_{ni} \propto \beta_{iw_n} exp\{\mathbb{E}_q[log(\theta_i)|\gamma]\} \tag{2}$$

$$\gamma_i = \alpha_i + \sum_{n=1}^{N} \phi_{ni} \tag{3}$$

, where it is shown in A.1 of Blei et al. [5] that:

$$\mathbb{E}_q[log(\theta_i)|\gamma] = \Psi(\gamma_i) - \Psi(\sum_{j=1}^{k} \gamma_j)$$

,where Ψ is the first derivative of the logΓ function.

Intuitively, Eq. 3 updates γ, which is the posterior Dirichlet, given the expected observation under the variation distribution, $\mathbb{E}[z_n|\phi_n]$. In the context of document modelling, γ_i is the posterior updated belief of the distribution of document i over the topics $\mathbf{z}$, which depends on values of β for words which appear in the document i and a *prior* belief of the distribution of the document over the topics. Hence, the updates in equation 2 and 3 are iterated over until convergence of γ_i.

As shown, the variation parameters $\{\gamma, \phi_1, ...\phi_N\}$ can be updated if α and β are known. However, α and β are the parameters we wish to infer. That is, we wish to find α and β that maximize the log likelihood of the data:

$$(\alpha^*, \beta^*) = \text{argmax}_{\alpha,\beta} \sum_{d=1}^{M} \log\, p(\mathbf{w}_d|\alpha, \beta). \tag{4}$$

Hence we use the Expectation-Maximization(EM) algorithm.

2.1.2 EM Parameter Estimation

We use the EM algorithm to find (α^*, β^*):

- In the E-Step, we use the iterative update steps in Eq. 2 and 3 to find the optimal values of the variational parameters γ_d^* and ϕ_d^* for every document.
- In the M-Step, we maximize the lower bound on the log likelihood, as shown in Eq. 4. As shown in A.3 and A.4 of Blei et al. (2003), we can find the maximum likelihood estimates of α and β using the below equations:

1. $\beta_{ij} \propto \sum_{d=1}^{M} \sum_{n=1}^{N_d} \phi^{*}_{ni} w^{j}_{dn}$
2. α requires a linear-scaling Newton-Rhapson algorithm to determine the optimal alpha:

$$\log(\alpha^{t+1}) = \log(\alpha^{t}) - \frac{\frac{dL}{d\alpha}}{\frac{d^2L}{d\alpha^2}\alpha + \frac{dL}{d\alpha}}$$

,where

$$\frac{dL}{d\alpha} = M(k\Psi'(k\alpha) - k\Psi'(\alpha)) + \sum_{d=1}^{M}(\Psi(\gamma_{di} - \Psi(\sum_{j=1}^{k}\gamma_{dj})))$$

$$\frac{d^2L}{d\alpha^2} = M(k^2\Psi''(k\alpha) - k\Psi''(\alpha))$$

By iterating over the E and M steps until convergence of the log likelihood, we obtain approximate empirical Bayes estimates for the LDA model.

2.2 Comparison to Other Models

We compare LDA to a Mixture of Unigrams and probabilistic latent semantic indexing (pLSI). We identify the shortcomings of these models for topic modeling and explain how LDA accounts for them. In particular, we address how each model handles documents being generated from (multiple) topics and how the models generalize to unseen data. Figure 3 shows the graphical representation of the models described in this section.

2.2.1 Continuous Mixture of Unigrams

In the unigram model, the words of every document are drawn independently from a single multinomial distribution:

$$p(\mathbf{w}) = \prod_{n=1}^{N} p(w_n)$$

If we augment this model with a discrete random variable z, the topic variable, we have a mixture of unigrams. In this mixture model, each document is generated by first choosing *exactly one* topic z and generating N words independently from $p(w|z)$, a conditional multinomial. The probability of a document is:

$$p(\mathbf{w}) = \sum_{z} p(z) \prod_{n=1}^{N} p(w_n|z)$$

We show in our empirical results (collaborative filtering) that the assumption that each document comes from exactly one topic is too limiting and that the model does not generalize as well to unseen data when compared to LDA, which allows documents to exhibit multiple topics at varying weights

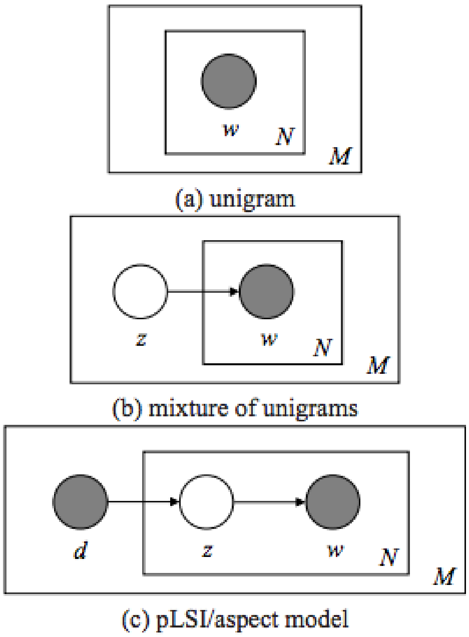
2.2.2 Probabilistic Latent Semantic Indexing

Introduced in 1999 by [12], pLSI assumes that a document label d and a word w_n are conditionally independent given a latent topic z:

$$p(d, w_n) = p(d) \sum_{z} p(w_n|z)p(z|d)$$

This model relaxes the assumption of a document being generated from only one topic that the mixture of unigrams makes. While pLSI does account for the fact that a document may contain multiple topics with $p(z|d)$ serving as mixture weight of topics for a particular document, this document index d is only defined over the training set. So, d, a multinomial random variable, only has as many possible values as there are training documents. Since there is no natural way to assign probability to unseen documents, pLSI is not a well-defined generative model of documents. Furthermore, a k-topic pLSI model has k multinomial distributions of size V and M mixtures over the k hidden topics. So, the $kV + kM$ parameters grows linearly with M. Because of this, pLSI can overfit to training data even when a regularization parameter is introduced to smooth the model [15].

LDA addresses these issues by treating the topic mixture weight as a k-parameter *hidden* random variable, whereas pLSI directly links topic parameters directly to training data. So, LDA's $k + kV$ parameters does not grow linearly with the number of documents in the corpus. LDA is also a well-defined generative model so it generalizes to new, unseen documents.



(a) unigram

(b) mixture of unigrams

(c) pLSI/aspect model

Figure 3: Graphical model representation of other models for discrete data. From [5].

3. EMPIRICAL RESULTS

We implement the LDA algorithm,as described in section 2.1.2, in Matlab. In this section, we attempt to replicate and extend 2 of the 3 empirical examples described in Blei et al. (2003). In particular, we apply LDA to the two problems of feature reduction for document classification and measuring generalizability (perplexity) for collaborative filtering.

3.1 Document Classification

In document classification, we often wish to classify a document into two or more mutually exclusive classes. A challenge faced in this problem is the choice of features. Treating individual words as features gives a large set of features and requires a huge amount of storage and lengthens runtime. One method of reduce the feature set is to use an LDA model for dimensionality reduction. Essentially we use the posterior $\gamma^*(\mathbf{w})$ values associated with each document in the corpus to reduce the features. In this context, instead of using all the words as features, we use our posterior belief of the distribution of documents over the topics.

Similar to the original example in Blei et al (2003), we use the Reuters-21578 dataset [13]. In this experiment, we estimate the parameters of an LDA model on all the documents without reference to their true class label. Then we run SVM on the feature set with reduced dimensions and compare it to an SVM trained on all the word features. While we use our own implementation of LDA, we use the builtin *fitcsvm* function in Matlab for the comparison.

First, we attempt to replicate the results in 7.2 of Blei et al (2003). However, we note several discrepancies with the dataset used for our experiment. While we found the full Reuters-21578 dataset online, the original files are in sgm files in a markup format difficult to parse. Instead, we found a Matlab dataset provided by Cai et al [6] that preprocesses the original dataset by discarding documents with multiple categories. Interestingly, while Blei et al [5] claimed to use a dataset with 8000 documents and 15,818 words, our dataset after preprocessing had 8293 documents and 18,933 words.

3.1.1 Comparing Results

To compare performance, we similarly use a 50-topic model for LDA. This represents a 99.7% reduction in the feature space. Figures 4 and 5 show our results. We see that there is little reduction in classification performance in using the LDA-based features, similar to those in Blei et al [5], shown for convenience in Figures 6 and 7. Note that we show results on ACQ vs NOT ACQ (Figure 5) and not GRAIN vs. NOT GRAIN (Figure 7) because our preprocessed dataset, which discarded documents with multiple categories, had too few documents with GRAIN to obtain meaningful results. We hypothesise that the original experiment kept all the categories that documents belonged to.

The slight differences between Figure 4 and Figure 6 could be explained by several factors. First, it is unclear the datasets received the same pre-processing, such as removal of stop words or removal of documents. Second, the resulting $\gamma^*(\mathbf{w})$ is a local optimum which depends on the initialization parameters. Third, there is inherent differences built into the randomness of splitting the data into training and testing data. Out of these possible reasons, we suspect the one which caused the greatest divergence in results is the first, since if the authors had kept all documents including ones with multiple categories, they would have potentially many more documents that contained EARN.

To extend the results slightly, we also show the performance after transforming the document-frequency dataset using the commonly used tf-idf technique [2]. This simply generates document vectors of the form $(1+\log(tf))*\log(\frac{N}{df})$. We find that while this feature set out performed the full word-features slightly in some cases, the improvement was marginal.

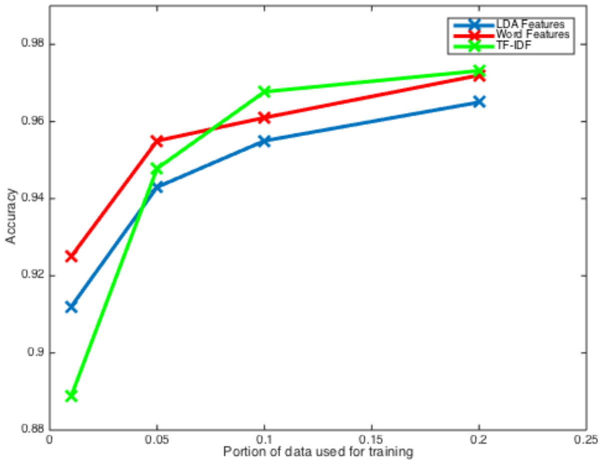


Figure 4: Classification results for EARN vs. NOT EARN binary classification from Reuters-21578 dataset for different proportions of training data

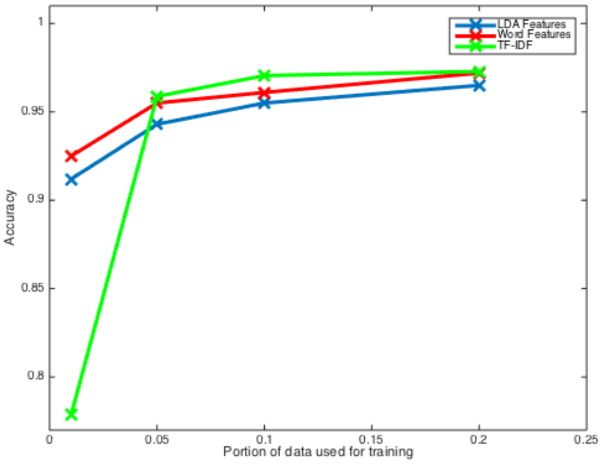


Figure 5: Classification results for ACQ vs. NOT ACQ binary classification from Reuters-21578 dataset for different proportions of training data

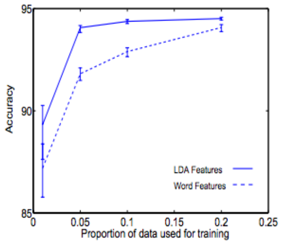


Figure 6: Classification results for EARN vs. NOT EARN in original experiment by Blei et al. (2003)

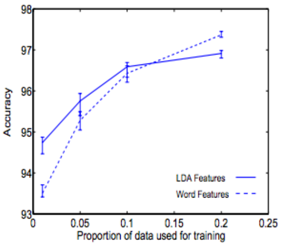


Figure 7: Classification results for GRAIN vs. NOT GRAIN in original experiment by Blei et al. (2003)

3.1.2 Extension

One question that arises from the original experiment is how the authors picked a 50-topic model for LDA. In particular, if one were to use LDA to reduce their feature set, what is the appropriate number of topics, K, to choose. We explore this extension by cross-validation to see what the optimal number of topics is for this dataset.

Figure 8 shows how the average accuracy of a cross validated SVM varies as we change the number of topics. We find that 25 topics performs the best in this example.

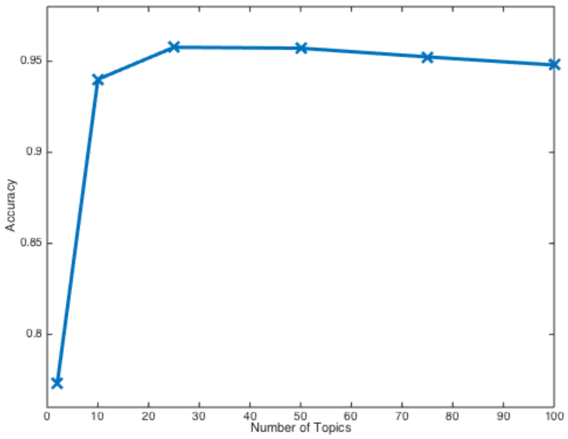


Figure 8: Average accuracy of SVM when trained on 10% of data, varying the number of topics used in LDA for feature reduction

3.2 Collaborative Filtering

The second problem is to see how well LDA performs in collaborative filtering. Using movie data, whereby users who indicate their preferred movie choices is analogous to a document with the words in it, we train the LDA model on a fully observed set of users. Then, for each unobserved user, we are shown all but one of the movies preferred by that user and asked how well LDA predicts what the held-out movie is. In particular, we use the notion of predictive-perplexity (PPL) as an evaluation metric of the likelihood of the held-out movie:

$$PPL(D_{test}) = exp\{-\frac{\sum_{d=1}^{M}\log p(w_{d,N_d}|\mathbf{w}_{d,1:N_d}-)}{M}\}$$

In the original paper [5], the authors compared the PPL of LDA against Fold in pLSI and Smoothed Mixt. Unigrams. As the implementation of the other algorithms is outside the scope of this paper, we use a pLSI algorithm found online for comparison. We could not find an online implementation of smoothed Mixture Unigrams so we implemented this algorithm ourselves.

3.2.1 Dataset

While the original experiment by Blei et al [5] used the EachMovie dataset, that dataset is no longer available and has been replaced by a dataset called MovieLens [1]. In the original dataset, the authors had a total of 3690 users and 1600 movies after restricting the EachMovie dataset to users that positively rated at least 100 movies (a positive rating is at least 4 out of 5 stars).

In the MovieLens dataset, we begin with 72,000 users and 10,000 movies. In an attempt to reduce the number of movies down to 1600 for closer comparison and faster runtime, we restrict the dataset to the most occurring top 1600 movies. From this, we further restrict the dataset to users that positively rated at least 100 movies and we obtain 12,035 users and 1600 movies. Since we wish to make use the same number of training and test users as the original experiment, we randomly select 3690 users from this dataset for our experiments.

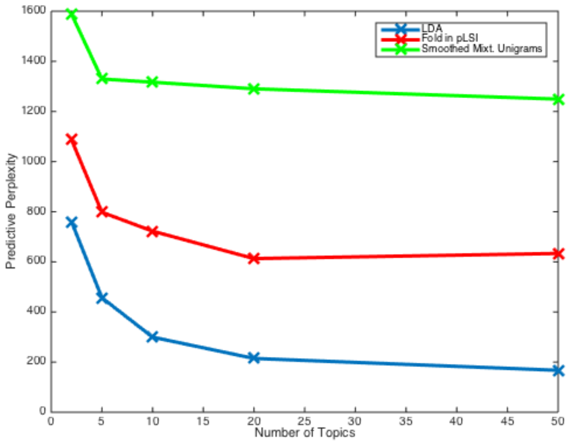


Figure 9: Results for collaborative filtering on the MovieLens data, varying the number of topics

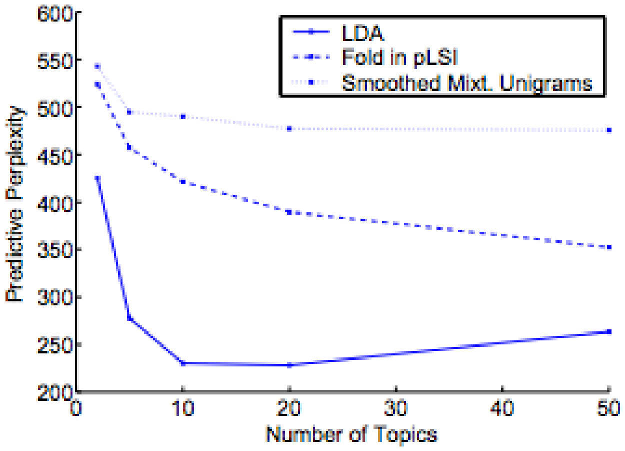


Figure 10: Results for collaborative filtering on the EachMovie data from the original paper by Blei et al. (2003), varying the number of topics

3.2.2 Comparing Results

By varying the number of topics used, we find the predictive-perplexities illustrated in Figure 9. For comparison, we show the results from the original experiment from [5] in Figure 10.

As shown, we find that the predictive-perplexity is much lower when using LDA compared to using Fold-in pLSI and Smoothed Mixture Unigrams. There are many reasons why the magnitude of the predictive perplexity is different in Figures 9 and 10. Perhaps the main one is that by using an essentially different dataset, we make a number of aforementioned assumptions on how to scale our dataset down to a comparable size and scope. Hence, our filtering could have reduced our dataset to exhibit different amounts of similarity between user's preferences in the corpus. Thus, with users less similar to each other, we expect to see our predictive perplexity to be generally larger than those reported by [5]. Also, our smoothing technique to correct for overfitting in both pLSI and Mixture of Unigrams could be different, resulting in higher PPL.

4. DISCUSSION

We address our findings and discoveries in the context of the state of the art. While it is common to refer to ”words” and ”documents” with LDA, it is important to note that LDA and similar topic models extend to discrete data in general, not just text data. LDA has seen recent use in the fields of bioinformatics and computer vision.

4.1 Bag of Words Assumption

The bag of words assumption that the order of the words does not matter and the words are therefore exchangeable is critical to the models we discuss. This assumption enables efficient models as we can ignore complex factors such as grammar and word order. However, this consideration of words independent of its context has limitations, such as the same word having multiple meanings [9].

4.2 Selecting Number of Topics

Choosing K, the number of topics, is a standard model selection problem. In 3.1.2, we used cross-validation to attempt to choose the K. Other methods include using annealed importance sampling to approximate the evidence, using the variational lower bound as a proxy for log $p(D|K)$ or using non-parametric Bayesian methods [14].

4.3 Further Application of LDA

The LDA model is modular, enabling extensibility. We list a few extensions of LDA:

- *Correlated Topic Model.* CTM address the problem of learning topic correlations from data. The Dirichlet distribution to model variability among topic proportions makes LDA unable to model topic correlation. The CTM model uses a logistic normal distribution to model topic proportions to assess topic correlation [4].

- *Hierarchical LDA.* hLDA addresses the problem of learning topic hierarchies from data. A nonparametric prior (referred to as the ”nested Chinese restaurant process”) is combined with a likelihood based on a hierarchical variant of LDA to create a hierarchical topic model [11].

- *Spatial LDA.* SLDA addresses the problem of modeling spatial and temporal structure among visual ”words.” The assignment of words to a document becomes a latent variable so that a generative procedure can group visual words which are close in space into the same document. Applications for SLDA include computer vision [18].

4.4 Relating to Blei 2003

Some ambiguities arise in [5] and we made assumptions or judgement calls as a result. The experiments compare performance between different models, so the ambiguities do not weaken the comparisons between models. They do make exact replication of the experiments challenging, however.

For the feature reduction for document classification, the exact details of SVM were omitted. We used the MATLAB implementation of SVM with a RBF kernel for our analysis.

We choose to use different smoothing techniques than [5]. For pLSI, we used a Laplacian smoothing [8]. For mixture of unigrams, we used a maximum-likelihood model over the entire corpus and then normalized. The potentially different smoothing techniques results in the exact perplexity values between our experiments and those in [5] impossible. Nevertheless, in both experiments, the perplexity of LDA was lesser than pLSI which was lesser than the smoothed mixture of unigrams model, so the results from [5] are verified.

We find that there are some details omitted that do not take away from the paper itself, but do make replication of the experiments more challenging. A common technique to address this now is to have a project website or repository available with code and datasets to make replication possible.

5. PROJECT DETAILS

5.1 Collaboration

We ensured work was distributed equally. Leung focused on the implementation of LDA and visualizing results. Xie worked on setting up the experiments by acquiring and cleaning the data and implementing the smooth mixture of unigrams and integrating pLSI. We both worked on the writeup.

5.2 Implementation

- We implemented the Variational Expectation-Maximization (EM) algorithm used for LDA.
- We implemented the calculation for perplexity, referring to [10], [14], [16].
- We implemented smooth mixture of unigrams, referring to [3], [14], [17].
- We implemented TF-IDF.
- We used the MATLAB implementation of SVM.
- We used a package for fold-in pLSI. [7].

6. REFERENCES

[1] Movielens. `http://grouplens.org/datasets/movielens/`.

[2] Tf-idf. `http://www.tfidf.com/`.

[3] Rafael E. Banchs. *Text Mining with MATLAB.* Springer-Verlag, 2013.

[4] David Blei and John Lafferty. Correlated topic models. *Advances in neural information processing systems*, 2006.

[5] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 2003.

[6] Deng Cai. Text datasets in matlab format. `http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html`.

[7] Deng Cai. Topic modeling and gmm on manifold (network). `http://www.cad.zju.edu.cn/home/dengcai/Data/LapPLSA.html`.

[8] Deng Cai, Qiaozhu Mei, Jiawei Han, and Chengxiang Zhai. Modeling hidden topics on document manifold. In *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 2008.

[9] Steven P. Crain, Ke Zhou, Shuang-Hong Yang, and Hongyuan Zha. Dimensionality reduction and topic modeling: From latent semantic indexing to latent dirichlet allocation and beyond. In *Mining Text Data*, pages 129–161. Springer US, 2012.

[10] Peter Gehler. Peter's code and dataset page. `http://people.kyb.tuebingen.mpg.de/pgehler/code/index.html`.

[11] DMBTL Griffiths and MIJJB Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. *Advances in neural information processing systems*, 2004.

[12] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1999.

[13] David D. Lewis. Reuters-21578. `http://www.daviddlewis.com/resources/testcollections/reuters21578/`, January 2012.

[14] Kevin Patrick Murphy. *Machine Learning: a Probabilistic Perspective*. MIT Press, Cambridge, MA, 2012.

[15] Alexandrin Popescul, Lyle H. Ungar, David M. Pennock, and Steve Lawrence. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, UAI '01, 2001.

[16] Colorado Reed. Latent dirichlet allocation: Towards a deeper understanding. January 2012.

[17] Allen B. Riddell. A simple topic model (mixture of unigrams). `https://ariddell.org/simple-topic-model.html`, July 2012.

[18] Xiaogang Wang and Eric Grimson. Spatial latent dirichlet allocation. In *Advances in neural information processing systems*, 2008.