Computational Machine Learning Final Project Practical Topic Modeling [*]

Viswanath Kammula(vk865@nyu.edu), Ke Ye(ky822@nyu.edu)

Abstract. The aim of this project is to build a topic modeling algorithm which uses maximum likelihood objective to derive the distribution of topics on a given set of vocabulary. We will try to improve on the existing probabilistic topic models and vector space models. The approaches used for topic model learning are either provable or practical but not both, we will try to present an algorithm for topic model learning which can be both practical and provable. The model we are developing will involve unsupervised learning techniques using Bayes rule. We will evaluate the learning model by using it for coherence as the performance metric. We will use NIPs and NY times datasets for the purpose of topic modeling.

1. Introduction.

Topic modeling is the method used to associate a given word with a particular topic, this can be mainly achieved through maximum likelihood objective. The model is simple: documents are mixtures of topics, which are modeled as distributions over a vocabulary. Each word token is generated by selecting a topic from a document-specific distribution, and then selecting a specific word from that topic-specific distribution.Recent work in theoretical computer science focuses on designing provably polynomial-time algorithms for topic modeling. We implement an algorithm that provably recovers the parameters of topic models provided that every topic contains at least one anchor word that has non-zero probability only in that topic. If a document contains this anchor word, then it is guaranteed that the corresponding topic is among the set of topics used to generate the document. The Algorithm has two steps,

- Selection Step - Selecting Anchor words for each topic.
- Recovery Step - Reconstructing topics distribution on selected anchor words.

Though standard topic modeling algorithms like LDA assume that topics are uncorrelated,there is a strong evidence that topics can co-occur. Economics and politics are more likely to co-occur than economics and cooking.we present a combinatorial anchor selection algorithm that does not require solving linear programs. So long as the separability assumption holds, we prove that this algorithm is stable in the presence of noise and thus has polynomial sample complexity, running in seconds on very large data sets.The important part of the algorithm is the recovery step,which computes topic-word distributions using matrix inversions. This step, which does noticeably affect performance, is sensitive to noise (whether from sampling or from model lack-of-fit) and so has high sample complexity and poor results even on synthetic data. We present a simple probabilistic interpretation of the recovery step that replaces matrix inversion with gradient-based inference. Finally to evaluate the performance of the model, we calculate the Coherence as our performance metric. We will first calculate the coherence for each topic over given distribution of words,and then we take the average coherence to evaluate

[*] This work was supported by Prof. Zaid Harchaoui(zaid.harchaoui@nyu.edu)

a model.

2. Data Acquisition and Processing. We have performed our modeling on NIPS(Neural Information processing systems) and New York Times dataset. We have collected the data from the below website. `http://archive.ics.uci.edu/ml/machine-learning-databases/bag-of-words`

The data which we acquired from UCI is formatted as follows, For each text collection, D is the number of documents, W is the number of words in the vocabulary, and N is the total number of words in the collection. After tokenization and removal of stopwords, the vocabulary of unique words was truncated by only keeping words that occurred more than fifty times. These data sets have no class labels, and for copyright reasons no filenames or other document-level metadata. These data sets are ideal for clustering and topic modeling experiments. Below are the statistics of the two datasets which we used,

Table 1
Information of the Datasets

Information	**NIPS**	**New York Times**
Number of Documents	1500	300000
Size of Vocabulary	12419	102626
Number of Words	1,900,00(approx)	100,000,00(approx)

We convert the data into sparse matrix format, we use CSR(compressed sparse row matrix) matrix format to perform our operations on our data. After truncation of the data we convert the word to document matrix into a word to word co-occurrence matrix called as Q matrix. We use this Q matrix as our main input for our anchor word selection algorithm.

3. Baseline Method. As part of the baseline we will implementing the Latent Dirichlet Allocation(LDA) for the purpose of topic modeling. LDA is an unsupervised learning algorithm which will try to group all the similar documents under a single topic. LDA is a generative probabilistic model for collections of discrete data such as text corpora. The goal is to find short descriptions of the members of a collection that enable efficient processing of large collections while preserving the essential statistical relationships.

LDA assumes the following generative process for each document w in a corpus D:

- $Choose\ N \sim Poisson(\xi)$
- $Choose\ \theta \sim Dir(\alpha)$
- For each of the N words we choose a topic $T_n \sim Multinomial(\theta)$.
- Then choose a word from $p(\xi_n|T_n, \beta)$, a multinomial probability conditioned on topic T_n.

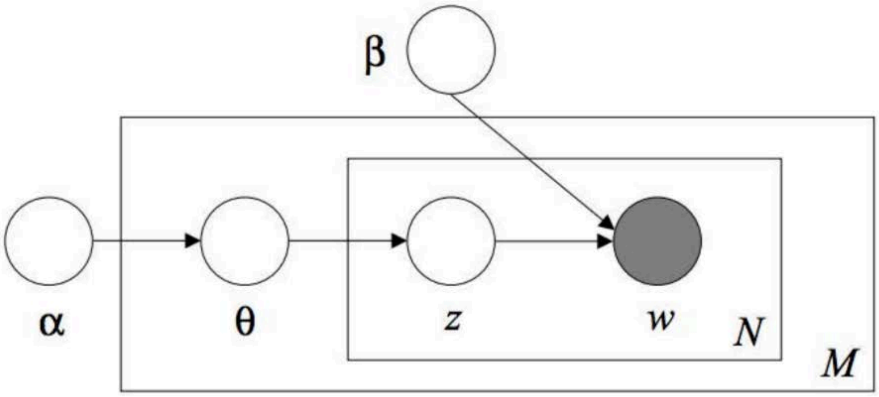

Figure 1. *The Probabilistic Graphical Model of LDA*

Figure 1 shows the probabilistic graphical model of LDA introduced by the paper of LDA(Blei et al., 2003), According to the paper of LDA (Blei et al., 2003), the probability of a sequence of words and topics can be expressed as

$$p(\xi, T) = \int p(\theta) \left(\prod_{n=1}^{N} p(T_n|\theta) p(\xi_n|T_n) \right) d\theta,$$

where θ is the parameter of a multinomial over topics. Thus, LDA will learn a conditional probability distribution $p(\xi|T)$ as the probability of the word ξ appears in the topic T for every latent topic T. We can then obtain the LDA distribution on documents by marginalizing out the topic variables with a Dirichlet distribution. We will get a k-dimensional vector by training k-topic model and then return the $p(\xi|T)$ values to the matrix. The output is a word-topic matrix which using the rows to represent word meanings. However, according to some previous research, we will see that LDA can't deliver robust word vectors. Therefore, we'd like to apply some new model to improve that as some later steps.

For evaluation of the model and also to find out how many topics exists in this corpus implicitly,We ran the model LDA model by increasing the number of topics for each run from 1 to 150 topics. And we also zoomed in and ran another series of models with number of topics from 1 to 30 to see if there is any subtle change inside. We calculated the coherence value for each run, which shows the performance of the model. When we plotted the Coherence values with the number of topics, we could observe that the coherence values getting decreased as the topics increase,and it started to converge when the number of topics was around 100.Coherence metric has been shown to correlate well with human judgments of topic quality. If we perfectly reconstruct topics, all the high-probability words in a topic should co-occur frequently, otherwise, the model may be mixing unrelated concepts. Given a set of words W, coherence is

$$Coherence(W) = \sum_{w_1, w_2 \in W} log \frac{D(w_1, w_2) + \epsilon}{D(w_2}$$

We used both the per-word perplexity and the Coherence to evaluate our model and plotted the trends of the performance with changing number of topics in several LDA models. Figure 2 is the trend of the Coherence while Figure 3 is the trend of the perplexity. We also collected all the generated topics in a single txt file, which was easy to follow the track of training. The Coherence and perplexity keep decreasing as the number of topics increased,As mentioned above, we've known that the less the perplexity, the better the performance of the model, so we can see the performance of the model increased as the number of topics increased, which did make sense. But when the number of topics reached 90, the result began to converge and change little afterwards. And the interesting is, the original assigned number of topics for this dataset is 90, we will dive deep into that using other improved models other than LDA and make some comparisons in our following work.

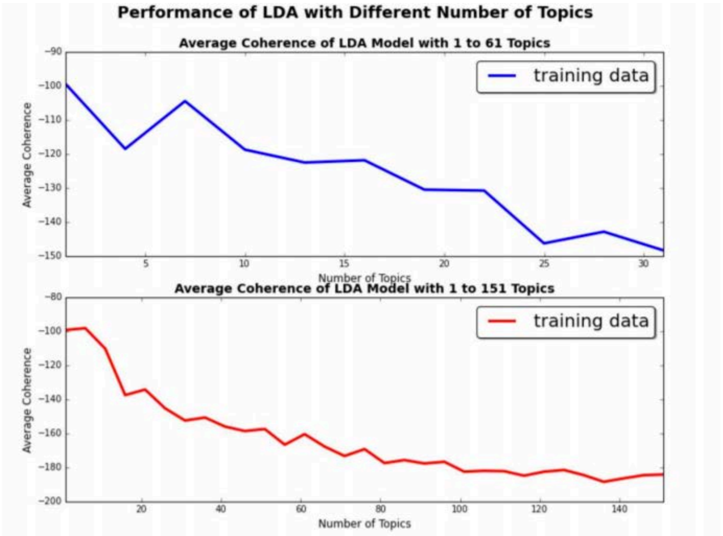

Figure 2. *The Coherence of the LDA models*

4. Methods and Techniques.

4.1. Anchor Words Selection. This is an algorithm that provably learns the parameters of a topic model given samples from the model, provided that the word-topic distributions are separable. According the definition in Arora Sanjeev et al(2012), The word-topic matrix A is p-separable for $p > 0$ if for each topic k, there is some word i such that $A_{i,k} \geq p$ and $A_{i,k'} = 0$ for $k^{'} \neq k$. When a word occurs in a document, it is defined as an anchor word. An anchor word is a perfect indicator that the document is at least partially about the corresponding topic, since there is no other topic that could have generated the word(Arora Sanjeev et al, 2012).

The Anchor words selection method takes the word to word co-occurrence matrix Q and the the total number of topics K as inputs. It starts by performing row normalization on Q matrix, so that the all the rows belonging to a single word sums up to one. This algorithm requires gaussian random projection of the Q-matrix to reduce its dimensionality. We first set the random_state as part of this projection and we set the new dimension to a fixed number which is 1000 in our case. We use the scikit-learn's gaussian random projection package as part of this step. The random matrix R can be generated using Gaussian distribution like this: The first row is a random unit vector uniformly chosen from S^{N-1}. The The second row is a random unit vector from the space orthogonal to the first row, the third row is a random unit vector from the space orthogonal to the first two rows, and so on.In this way of choosing R, the following properties can be satisfied:

- Spherical symmetry: For any orthogonal matrix $A \in O(N)$, R_A and R have the same distribution.
- Orthogonality: The rows of R are orthogonal to each other.
- Normality: The rows of R are unit-length vectors.

After projection the approach is to iteratively find the farthest point from the subspace spanned by the anchor words found so far.

- We first start by finding the farthest point from origin as our first anchor word.We will be using the dot product between the two points to calculate the distance between them.
- For i=1 to i=k-1, find the points that has the largest distance to span(S), which is the largest norm of the projection of x onto the orthogonal complement of span(S).

In real world scenario where we have infinitely many documents, the space of points of the rows in Q will be a simplex where the vertices of this simplex correspond to the anchor words. Given a set of points whose set P is a simplex, we wish to find the vertices of P. When we have a finite number of documents, the rows of Q are only an approximation to their expectation: we are given a set of points $d_1, d_2, ..., d_V$ that are each a perturbation of $a_1, a_2, ..., a_V$ whose space P defines a simplex. We would like to find an approximation to the vertices of P.

The main technical step is to show that if one has already found r points S that are close to r (distinct) anchor words, then the point that is farthest from span(S) will also be close to a (new) anchor word. Thus the procedure terminates with one point close to each anchor word, even in the presence of noise. A practical advantage of this procedure is that when faced with many choices for a next anchor word to find, our algorithm tends to find the one that is most different from the ones we have found so far. The algorithm terminates when it has found K anchors. K is a tunable parameter of the overall algorithm that determines how many topics are fitted to the dataset. Below are the set of Anchor words which were generated using different values of k,

- Anchor words for k=10: neuroscience, part, iii, cognitive, paul, science, architecture, theory, visual, processing
- Anchor words for k=50: neuroscience, part, iii, cognitive, michael, science, architec-

ture, theory, speech, processing, spike, visual, character, motion, control, classifier, face, polynomial, chip, action, view, word, motor, member, loss, orientation, expert, kernel, user, concept, recurrent, rotation, channel, inverse, synapses, tree, rules, cluster, winner, robot, hopfield, operator, display, posterior, oscillation, regular, matching, eye, mutual, trajectory

- Anchor words for k=80: neuroscience, part, iii, cognitive, paul, science, architecture, theory, visual, processing, policy, head, speaker, spike, character, motion, speech, face, chip, classifier, control, boolean, teacher, missing, sound, controller, view, kernel, word, orientation, expert, batch, motor, clustering, winner, hopfield, channel, convex, member, rules, letter, user, carlo, penalty, robot, contour, operator, synapses, loss, rbf, tree, rotation, separation, concept, taylor, light, ensemble, string, mutual, extra, interpolation, trajectory, oscillation, eye, content, building, module, matching, inverse, processor, codes, attention, subspace, capacity, tractable, weighting, utility, percepton, sensor, predictor

We can clearly see,that the anchor words from k=50 are same as words in k=10 with few more extra words,similarly with k=80 and k=50.

4.2. KL and L2 Recovery Method. We used this two new methods based on Bayesŕule. Considering any two words in a document w_1 and w_2 with corresponding topics z_1 and z_2, we used to index the matrix of word-topic distribution. Given a word2word matrix Q, its elements can be interpreted as

$$Q_{i,j} = p(w_1 = i, w_2 = j)$$

We denoted the row-normalized Q matrix as $Q^{'}$, which is essential for us to do the recovery step, for it can be interpreted as a conditional probability,

$$Q^{'}_{i,j} = p(w_2 = j, |w_1 = i)$$

We recovered matrix A by using Bayesŕule:

$$p(w_1 = i|z_1 = k) = \frac{p(z_1 = k|w_1 = i)p(w_1 = i)}{\sum_{i'} p(z_1 = k|w_1 = i')p(w_1 = i')}$$

We can solve for $p(w_1 = i)$ since $\sum_j Q_{i,j} = \sum_j p(w_1 = i, w_2 = j) = p(w_1 = i)$. The algorithm finds the vector of non-negative coefficients $p(z_1|w_1 = i)$ which best reconstruct each row of the empirical row normalized matrix $Q^{'}$ as a convex combination of the rows. The KL method use KL divergence as the objective function, while L2 method use quadratic loss as the objective to find the best reconstruct.

5. Evaluation Metric. If we perfectly reconstruct topics, all the high-probability words in a topic should co-occur frequently, otherwise, the model may be mixing unrelated concepts. The coherence is given by below equation,

$$Coherence(W) = \sum_{w_1,w_2 \in W} log \frac{D(w_1,w_2)+\epsilon}{D(w_2}$$

In order to calculate the coherence of the model, we first calculate coherence on each topic distribution. We consider a topic, and calculate the number of times the word occur with other words in a given document and divide it with the total number of document the word occurred in. We do this for all words that are recovered as part of distribution and get the sum of them as topic coherence. In order to get the coherence of the model, we take the average of all topic coherences. The performance of a topic modeling can be evaluated by calculating the average coherence coherence.The larger value of coherence refers to better performance. Since we are using log the values of coherence are found to be in negative.As part of our experiments we have calcuated the coherence of different Recovery methods(KL,L2 and Orginal Recovery) and LDA model.

6. Results.

We tried to implement three topic recovery methods based on anchor selection step: the original topic recovery method, a new recovery method using KL divergence as an objective loss and another recovery method using quadratic loss as objective loss. We tried to evaluate the performance of different recovery methods and compared these methods with our baseline method, the LDA model. Below are the tables 2,3,4, which show the anchor words which we recovered on three recovery methods which we used as part of the experiment for k=10.

Table 2
Original Recovery Method with K=10

Anchor words	Words Recovered using original Recovery Method
neuroscience	algorithm network input neural output unit function training system set
part	part neuron input system network direction function unit head response
iii	iii network function neural set error result system problem training
cognitive	cognitive network function unit set input data neural problem system
peter	david michael john richard peter thomas william andrew author index
science	science network unit representation set input weight level data problem
architecture	algorithm network input neural output unit training function system set
theory	theory network data spike set evidence function peak neuron neural
visual	visual network neural system input function data unit signal output
processing	neuron input system field cortex object orientation motion direction

We can see that the anchor words are the same for all three recovery methods except for some randomness. But the words recovered vary with the probability distributions. Figure 3,4 are the results showing the topic distributions using the KL and L2 divergence methods where we set the number of topics to be equal to 100.

Table 3
KL-Recovery Method with K=10

Anchor words	Words Recovered using KL-Recovery Method
neuroscience	implementation application navigation planning eric speech margin
part	part control planning navigation robot analog implementation action
iii	iii animal vowel linguistic psychological theories trigger perceived
cognitive	language representation activation grammar vowel semantic linguistic
paul	david michael john author index richard peter william paul thomas
science	action reinforcement policy environment agent robot reward exploration
architecture	network input neural unit algorithm training output set weight system
theory	function network algorithm set data problem result error parameter
visual	neuron visual field response direction motion system activity object
processing	signal network system input neural speech output information data

Table 4
L2 Recovery Method with K=10

Anchor words	Words Recovered using L2-Recovery Method
implementation	application navigation david richard thomas terrence robert michael
part	part network function input neural algorithm system set data training
iii	iii application david richard thomas robert terrence michael andrew
cognitive	application david planning richard robert michael terrence thomas
peter	network function input neural set system data training algorithm unit
science	application control algorithm navigation network language function
architecture	network algorithm architecture function input neural set data training
theory	network theory function input neural set algorithm data system
visual	network visual input function neural system set signal data neuron
processing	processing network input neural function system set data unit output

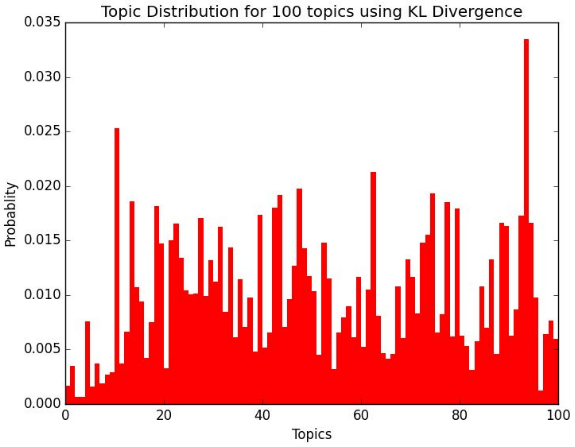

Figure 3. *Topic Distribution for 100 topics using KL Divergence*

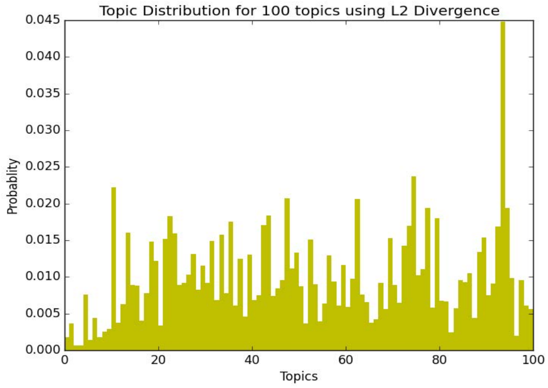

Figure 4. *Topic Distribution for 100 topics using L2 Divergence*

We can see a similar kind of distributions over the topics, but we find the probability of these varies. The topic which has highest probability shows that high proportion of documents in the corpus belongs to that particular topic. Next we find the probability of the topics using KL and L2 recovery, were we set k=10 and the number of words recovered to be 10. In Figure 5,6 below the topics are shown using the anchor word which represents the topic with there probabilities. Both models have the same set of anchor words but their probabilities are very different. We can evaluate which one of this is better only calculating the coherence values for two models.

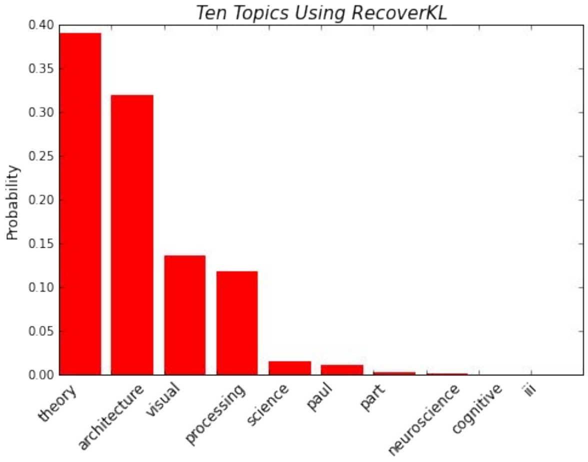

Figure 5. *Ten Topics Using RecoverKL*

We have evaluated the original recovery, L2 Recovery, KL recovery and LDA models using coherence as the performance metric, and we get the below, Figure 7, as results for these 4 models.

We have performed our experiment for k=20,100 and 150. We can clearly see for all values of K, we find that original recovery has least coherence value. We can find that KL and L2 have almost the same performances for k=50 and 100, But KL is better over L2 when K=150. It's

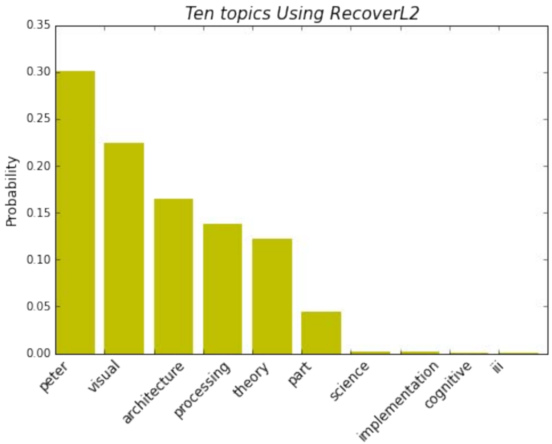

Figure 6. *Ten Topics Using RecoverL2*

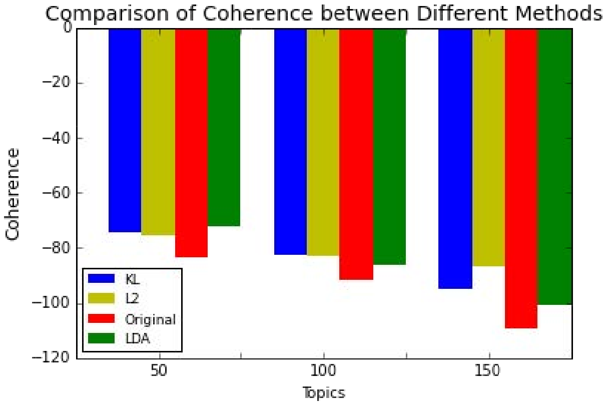

Figure 7. *Comparison of Coherence between Different Methods*

also evident from this method that the KL and L2 have performed well over the LDA model as the number of topics increases.

7. Conclusion. We've run our experiment on 4 models: Original Recovery, Recovery using L2 Divergence, Recovery using KL Divergence and LDA model. We have evaluated the performance and found that KL divergence has the best performance over our baseline method LDA. We conclude this on the basis that the anchor words methods are considering the correlation between the topics and the LDA model assumes the topics to independent. This model considers the correlation between the topics that are generated by running the model. The Coherence for the new recovery method is better compared to LDA and the origianal recovery method, while the KL Recovery method performs better than the other recovery methods. The running time of these algorithms is effectively independent of the size of the corpus

REFERENCES

[1] Learning word vectors for sentiment analysis., `http://ai.stanford.edu/~amaas/papers/wvSent_acl2011.pdf`

[2] A practical algorithm for topic modeling with proveable guarentees., `http://cs.nyu.edu/~dsontag/papers/AroraEtAl_icml13.pdf`

[3] Supplement material for A practical algorithm for topic modeling., `http://cs.nyu.edu/~dsontag/papers/AroraEtAl_icml13_supp.pdf`

[4] Latent Dirichlet Allocation., `https://www.cs.princeton.edu/~blei/papers/BleiNgJordan2003.pdf`

[5] Ailon, N. and Chazelle, B. Approximate nearest neighbors and the fast johnson-lindenstrauss transform. SICOMP, pp. 302322, 2009

[6] Anandkumar, A., Foster, D., Hsu, D., Kakade, S., and Liu, Y. Two svds suffice: Spectral decompositions for probabilistic topic modeling and latent dirichlet allocation. In NIPS, 2012

[7] Arora, S., Ge, R., Kannan, R., and Moitra, A. Computing a nonnegative matrix factorization provably. In STOC, pp. 145162, 2012a. 1, 4, 1

[8] Arora, S., Ge, R., and Moitra, A. Learning topic models going beyond svd. In FOCS, 2012b. 1, 2, 2, 2, 3, 2, 3, 1, 5

[9] Bittorf, V., Recht, B., Re, C., and Tropp, J. Factoring nonnegative matrices with linear programs. In NIPS, 2012. 1

[10] Blei, D. Introduction to probabilistic topic models. Communications of the ACM, pp. 7784, 2012. 1

[11] Blei, D. and Lafferty, J. A correlated topic model of science. Annals of Applied Statistics, pp. 1735, 2007. 1, 2

[12] Blei, D., Ng, A., and Jordan, M. Latent dirichlet allocation. Journal of Machine Learning Research, pp. 9931022, 2003. Preliminary version in NIPS 2001. 1, 2

[13] Buntine, Wray L. Estimating likelihoods for topic models. In Asian Conference on Machine Learning, 2009. 5.1

[14] Deerwester, S., Dumais, S., Landauer, T., Furnas, G., and Harshman, R. Indexing by latent semantic analysis. JASIS, pp. 391407, 1990. 1

[15] Donoho, D. and Stodden, V. When does non-negative matrix factorization give the correct decomposition into parts? In NIPS, 2003. 2

[16] Gillis, N. Robustness analysis of hotttopixx, a linear programming model for factoring nonnegative matrices, 2012. http://arxiv.org/abs/1211.6687. 1

[17] Gillis, N. and Vavasis, S. Fast and robust recursive algorithms for separable nonnegative matrix factorization, 2012. http://arxiv.org/abs/1208.1237. 1, 4

[18] Gomez, C., Borgne, H. Le, Allemand, P., Delacourt, C., and Ledru, P. N-findr method versus independent component analysis for lithological identification in hyperspectral imagery. Int. J. Remote Sens., 28(23), January 2007. 4

[19] Griffiths, T. L. and Steyvers, M. Finding scientific topics. Proceedings of the National Academy of Sciences, 101: 52285235, 2004. 1

[20] Kumar, A., Sindhwani, V., and Kambadur, P. Fast conical hull algorithms for near-separable non-negative matrix factorization. 2012. http://arxiv.org/abs/1210.1190v1. 4

[21] Li, W. and McCallum, A. Pachinko allocation: Dagstructured mixture models of topic correlations. In ICML, pp. 633640, 2007. 1, 2

[22] McCallum, A.K. Mallet: A machine learning for language toolkit, 2002. http://mallet.cs.umass.edu. 2, 5

[23] Mimno, David, Wallach, Hanna, Talley, Edmund, Leenders, Miriam, and McCallum, Andrew. Optimizing semantic coherence in topic models. In EMNLP, 2011. 5.1

[24] Nascimento, J.M. P. and Dias, J. M. B. Vertex component analysis: A fast algorithm to unmix hyperspectral data. IEEE TRANS. GEOSCI. REM. SENS, 43:898910, 2004. 4

[25] Stevens, Keith, Kegelmeyer, Philip, Andrzejewski, David, and Buttler, David. Exploring topic coherence over many models and many topics. In EMNLP, 2012. 5.1

[26] Thurau, C., Kersting, K., and Bauckhage, C. Yes we can simplex volume maximization for descriptive webscale matrix factorization. In CIKM10, 2010. 4

[27] Wallach, Hanna, Murray, Iain, Salakhutdinov, Ruslan, and Mimno, David. Evaluation methods for topic models. In ICML, 2009. 5.1

[28] Yao, Limin, Mimno, David, and McCallum, Andrew. Ef- ficient methods for topic model inference on streaming document collections. In KDD, 2009. 5.2