IBM IPAM RIPS 2010 Program
Topic Detection and Causal Influence in Microblogs

Student Team
Edward Chang (Yale University
Edward Dewey (Swarthmore College)
Seçin Şahin (Middle Eastern Technical University)
Maksim Tsikhanovich (Bard College)

Academic Mentor
Blake Hunter[*]
Department of Mathematics
University of California, Davis
blakehunter@math.ucdavis.edu

IBM Mentors
Dmitriy Katz-Roghozhnikov
Aurelie C. Lozano
Prem Melville
Vikas Sindhwani[†]

Business Analytics and Mathematical Sciences
IBM T.J. Watson Research Center
Yorktown Heights, NY
{vsindhw, pmelvil, aclozano, dkatzrog}@us.ibm.com

1 Introduction

The emergence of the web2.0 phenomenon has set in place a planetary-scale infrastructure for rapid proliferation of information and ideas. Social media platforms such as blogs, twitter accounts and online discussion sites are large-scale forums where every individual can voice a potentially influential public opinion. According to recent surveys, a massive number of internet users are turning to such forums to collect recommendations and reviews for products and services – shaping their choices and stances by the commentary of the online community as a whole.

This unprecedented scale of unstructured user-generated web content presents new challenges to both consumers and companies alike. Which blogs or twitter accounts should a consumer follow in order to get a gist of the community opinion as a whole? How can a company identify bloggers whose commentary can change brand perceptions across this universe, so that marketing interventions can be effectively strategized?

Microblogs are particularly interesting. The most successful microblogging site, Twitter, has shown exponential growth recently in terms of number of users and the number of tweets. It is rapidly becoming the first point of release of information surpassing traditional media. For example, the news of the 2009 landing of US Airways 1549 in the Hudson was first broken on twitter by a passenger. It is also representatitive of social media platforms that serve as records of large-scale,

[*] Academic Coordinator
[†] IBM Coordinator

instanteneous human commentary on world events. For this reason, Twitter will be archived by the US Library of Congress.

In this project, our testbed will be datasets drawn from Twitter and we will focus on two classes of mathematical modeling problems described below.

1. How can we essentially “compress” large number of tweets into a smaller collection of topics? By summarizing the entire collection of tweets pertaining to a domain, topics are easier to digest by a human analyst.
2. How can we characterize the flow of information, and in particular the spread of influence, in the community of microbloggers?

An overview of Machine learning techniques for social media analytics appears in [17]. See references therein for representative papers from IBM Research.

2 Data Generation

2.1 Twitter Domains

We have generated a collection of twitter datasets corresponding to different *domains* of interest. For example, some of these domains include different IBM product/business lines[1]. Each domain is specified by a collection of keywords. Using Twitter's search API [2], all tweets containing these domain-specific keywords over a time period are collected. Each tweet is associated with a timestamp and an account ID that uniquely identifies the author.

2.2 Text Representation

Each tweet may be viewed as a mini-document. We use the classic bag-of-words representation to turn tweets into numerical feature vectors. This involves two main steps: (a) Vocabulary generation and (b) Indexing. In the vocabulary generation step, tweets are tokenized by white space, common stop words are removed, words are converted into stemmed terms, and finally terms that occur in fewer than a certain number of documents are pruned away as a denoising heuristic. The resulting set of words form a vocabulary of d words. In the indexing step, each tweet is represented using TFIDF [16] feature vectors, $x = (x_1 \ldots x_d) \in \mathbf{R}^d$ with entries $x_j = tf_j * idf_j$ where tf_j is the term-frequency of the j^{th} term of the vocabulary in the tweet and idf_j is the associated *inverse document frequency* (IDF). IDF is defined as $idf(j) = ln(\frac{n}{df_j})$ where n is the total number of tweets in the collection and df_j is the number of documents containing term j.

The TFIDF representation is an effective heuristic developed in the information retrieval community. It emphasizes words that occur several times in a small collection of documents over highly common words that have less discriminatory power. The entire tweet collection may be represented as the document-term matrix $\mathbf{X} \in \mathbf{R}^{n \times d}$ whose rows represent n tweets in the collection represented over d words. It is further common to normalize the rows to have unit l_1 or l_2 norm so that each document is treated equally.

3 Detecting Topics with Sparse Low-rank Matrix Approximations

Topic models commonly operate under the assumption that it is possible to summarize a large collection of text documents in terms of a much smaller set of discussion themes. These discussion themes are identified as “topics” in the document collection. Topic modeling is closely related to clustering and as such may be viewed as “soft-clustering”. Two families of topic models have been explored in the literature. The first is based on Low-Rank Matrix Factorizations [16, 12, 21] while the second is based on Probabilistic Topic models [18, 10, 2]. These families are actually closely related as we explain below.

[1] e.g., Lotus `www.ibm.com/software/lotus`

[2] `http://apiwiki.twitter.com/Search-API-Documentation`

Let k be the number of topics typically specified as an input parameter. Each topic is associated with a (column) vector $h \in \mathbf{R}^d$ which in probabilistic topic models is a distribution over words or may be viewed as a member of a dictionary over which documents can be effectively coded. Let $\mathbf{H} \in \mathbf{R}^{k \times d} = (h_1 \dots h_k)^T$ represent the topic-word matrix. Then, we want each document (tweet) to be approximately expressible by linear mixing the collection of topics. Let x be a row-vector representing a document. Then, we want,

$$x \approx \sum_{i=1}^{k} w_i h_i' = w'\mathbf{H}$$

where w are the mixture coefficients. In other words, we want to express the document-term matrix $\mathbf{X}$ in terms of a *low-rank approximation*,

$$\mathbf{X} \approx \mathbf{WH}$$

where $\mathbf{W}$ is the $n \times k$ matrix of document-topic associations and $\mathbf{H}$ is $k \times d$ matrix of topic-term associations. A human analyst can interpret/explore the topic by examing the top words as ranked by h_r or the top documents as ranked by w_r. These are documents and words that associate most strongly with the topic.

We now quickly summarize some connections.

1. Let us measure the quality of approximation using the Frobenius norm. In other words, let us minimize,

   $$J(\mathbf{W}, \mathbf{H}) = \|\mathbf{X} - \mathbf{WH}\|^2$$

   Firstly, this problem does not have a unique solution in $\mathbf{W}$ and $\mathbf{H}$ since $\mathbf{WH} = \mathbf{WQQ}^{-1}\mathbf{H}$ for any invertible $k \times k$ matrix $\mathbf{Q}$. The best *rank-k* approximation to a matrix is given by truncating the SVD. Let $\mathbf{X} = \mathbf{USV}^T$ be the SVD of $\mathbf{X}$. Thus one solution to the problem above is $\mathbf{W} = \mathbf{U}_k\mathbf{S}_k^{\frac{1}{2}}$ and $\mathbf{H} = \mathbf{S}_k^{\frac{1}{2}}\mathbf{V}_k^T$ where the subscript k denotes restriction to the top k singular vectors and singular values. This approach is classically called *Latent Semantic Analysis* in the Information Retrieval literature where the singular vectors are used to find lower k-dimensional embeddings of documents and terms, and similarity computations for search applications are conducted in the “latent space” [16]. Also see [20] for connections to k-means clustering.

2. Let us now assume that $\mathbf{X}$ is normalized so that the sum of its entries is 1. Then $\mathbf{X}$ may be viewed as specifying a joint probability distribution over two random variables $\mathcal{D} \in \{1, \dots, n\}$ and $\mathcal{T} \in \{1, \dots, d\}$. The pLSA model of [10] was introduced as a probabilistic enhancement of LSI to better model discrete term-frequency counts. It assumes the existence of k hidden variables, $(z_1, \dots, z_k)$, with respect to which this joint distribution factorizes. In other words,

   $$P(\mathcal{D} = i, \mathcal{T} = j) = \sum_{t=1}^{k} P(z = t)P(\mathcal{D} = i|z = t)P(\mathcal{T} = j|z = t)$$

   Given the observations, $\mathbf{X}_{ij}$, we can now learn the parameters of the probabilistic model by maximizing the log-likelihood of the data,

   $$J(\{P(z = t), P(\mathcal{D} = \cdot|z = t)P(\mathcal{T} = \cdot|z = t)\}_{t=1}^{k}) = \sum_{ij} \mathbf{X}_{ij} \log(P(\mathcal{D} = i, \mathcal{T} = j))$$

   It turns out [6] that this is equivalent to minimizing the *Generalized KL-divergence* under *non-negativity* constraints $\mathbf{W} \geq 0, \mathbf{H} \geq 0$,

   $$J(\mathbf{W}, \mathbf{H}) = \sum_{ij} \mathbf{X}_{ij} \ \log\left(\frac{\mathbf{X}_{ij}}{(WH)_{ij}}\right) + (WH)_{ij} - \mathbf{X}_{ij}$$

   Non-negative Matrix Factorizations (NMF) refer to a class of techniques to build low-rank matrix approximations to minimize such divergence criteria under the constraint that the low-rank factors be non-negative. It was popularized by the work of Lee and Seung in 1999 [12] where the motivation was to interpret dictionary elements (rows of $\mathbf{H}$) as "parts" of an object that needed to be recovered from an object collection. For better interpretability of low-rank factorization of non-negative data, it is useful to disallow basis elements with negative entries and allow only additive combinations of them (i.e., $\mathbf{W} \geq 0$) to reconstitute objects. [12] and other followup papers give very simple update rules for learning an NMF. Note however, in contrast to the SVD, the NMF problem is NP-hard [19].

3. LDA [2] is a further enhancement of pLSI where priors are imposed over model parameters to avoid overfitting and to lend a better probabilistic interpretation to the model. By "better", we mean a generative model that says how an entire document is generated as opposed to document and word *indices* in pLSI. LSA is widely considered to be a state of the art methodology for topic modeling. A matlab toolbox for LDA is available[3].

   In an NMF framework, priors may be enforced via regularization terms. An example of a regularized objective function is the following,

$$J(\mathbf{W}, \mathbf{H}) = \|\mathbf{X} - \mathbf{WH}\|^2 + \gamma \|\mathbf{W}\|_1$$

   which enforces the "coding" to be sparse via the l_1 penalty. The l_1 penalty is a convex proxy to the l_0 norm that directly measures the number of non-zeros in $\mathbf{W}$. Sparsity is useful because it can lead to improved clustering effect and also allow the optimization algorithm to scale to larger number of documents and/or topics by maintaining each low-rank factor as a sparse (as opposed to dense) matrix.

4. NMF is closely related to several clustering algorithms including spectral clustering and k-means [5, 13].

3.1 Optimization Algorithms

There are a large number of optimization algorithms proposed for solving NMF problems. These include Lee and Seung's classic multiplicative update algorithms [12], Alternating Least Squares [1], Alternating Non-negative Least Squares and Projected Gradient Methods [14]. More algorithms are exhaustively covered in the book [4]. A recent elegant approach called *Rank-one Residue Iterations* was independently proposed by three authors [9, 7, 4]. It is based on the observation that a class of rank-one subproblems have the following closed form update rules under non-negativity constraints,

$$h_r = \arg\min_{h \geq 0} \|\mathbf{R} - w_r h^T\|^2 = \frac{1}{w_r' w_r} \max(\mathbf{R}' w_r, 0)$$

where $\mathbf{R} = (\mathbf{X} - \sum_{j \neq r} w_j h_j^T)$ is the current residual matrix (note: not necessarily non-negative). Since the product $\mathbf{WH}$ is not unique, it is common to normalize $\mathbf{WH} = (\mathbf{W} D_H)(D_H^{-1} \mathbf{H})$ where D_H is a diagonal matrix of row sums of $\mathbf{H}$. In other words, h_r is reset to $\frac{1}{1^T h_r} h_r$ while w_r is reset to $(1^T h_r) w_r$ without changing the objective value.

Very similar update rules can be derived for w_r:

$$w_r = \arg\min_{w \geq 0} \|\mathbf{R} - w h_r^T\|^2 = \frac{1}{h_r' h_r} \max(\mathbf{R} h_r, 0)$$

The algorithm cycles over the variables $\mathbf{W} = (w_1 \dots w_k)$ and $\mathbf{H} = (h_1 \dots h_k)^T$ and with slight modifications is guaranteed to converge to a stationary point of the objective function.

Let us consider the following objective function that enforces sparsity in $\mathbf{W}$:

$$J(\mathbf{W}, \mathbf{H}) = \|\mathbf{X} - \mathbf{WH}\|^2 + \gamma \|\mathbf{W}\|_1$$

Because of the scale invariance, let us add the following constraints:

[3] http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm

$$\mathbf{H}1 \leq 1 \quad or \quad \|h_r\|_1 \leq 1 \;\; r = 1, \ldots, k$$

Without such constraints, the algorithm can minimize the l_1 norm by scaling down $\epsilon\mathbf{W}$ and scaling up $\frac{1}{\epsilon}\mathbf{H}$ taking $\epsilon \rightarrow 0$.

When l_1 regularization is added, the update rules become,

$$w_r = \arg\min_{h \geq 0} \|\mathbf{R} - w_r h^T\|^2 + \lambda\|w_r\|_1 = \frac{1}{h_r' h_r} \max(\mathbf{R}h_r - \lambda, 0)$$

where λ shows up naturally as a thresholding operator.

The h_r update remains the same,

$$h_r = \arg\min_{h \geq 0} \|\mathbf{R} - w_r h^T\|^2 = \frac{1}{w_r' w_r} \max(\mathbf{R}' w_r, 0)$$

except that if $\|h_r\|_1 > 1$, we can (safetly) reset $h_r = \frac{1}{\|h_r\|_1} h_r$ and satisfy the constraints. Because h_r is in a descent direction and the objective function is quadratic, this resetting does not increase the objective value.

Note that enforcing sparsity in $\mathbf{W}$ also affects the sparsity in $\mathbf{H}$.

Some preliminary exploration of sparsity in the context of document clustering appears in [11].

3.2 Measuring Effect of Sparse Coding

We are interested in understanding whether sparsity leads to better topic models while significantly reducing storage requirements. One way to measure quality of a topic model is the reconstruction error on a held out test set,

$$\min_{\mathbf{W} \geq 0} \|\mathbf{X}_{test} - \mathbf{W}\mathbf{H}\|^2$$

This may be viewed as an *intrinsic* measure that can be computed by running rounds of $\mathbf{W}$ optimization keeping $\mathbf{H}$ fixed. It is important to note however that evaluation of topic models is an area of ongoing research. A recent paper [3] proposed methods to incorporate human judgement in evaluating topic models which may be considered as *extrinsic* evaluation measures. The same paper notes that intrinsic measures often do not correlate with extrinsic evaluation. Another method for extrinsic evaluation is to measure clustering quality [21] or evaluate classifiers using topic-based data representations [2] on labeled datasets where each document has been manually associated with a category (the assumption is that labeled categories correspond to topics).

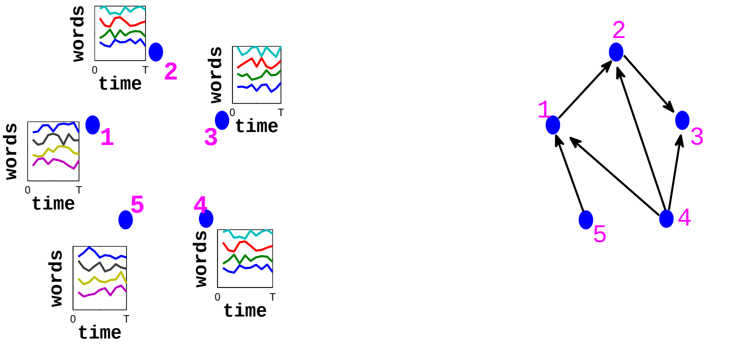
4 Temporal Causal Modeling for Characterizing Influence

The objective of this part of the project would be to understand how microbloggers influence each other. As an example of the mechanics of spread of influence, let us consider the following sequence of events. A consumer is looking to buy a laptop. She initiates a web search for the laptop model and browses several discussion and blog sites where that model has been reviewed. The reviews bring to her attention that among other nice features, the laptop also has excellent speaker quality. Next she buys the laptop and in a few days herself blogs about it. Arguably, conditional on being made aware of speaker quality in the reviews she had read, she is more likely to herself comment on that aspect without necessarily attempting to find those sites again in order to link to them in her blog. In other words, the actual post content is the only trace that the opinion was implicitly absorbed. Moreover, the temporal order of events in this interaction is indicative of the *direction of causal influence*.

It is possible to formulate these intuitions rigorously in terms of the notion of *Granger Causality* [8]. Introduced by the Nobel prize winning economist, Clive Granger, this notion has proven useful as an operational notion of causality in time series analysis. It is based on the intuition that a cause should necessarily precede its effect, and in particular if a time series variable X causally affects another Y, then the past values of X should be helpful in predicting the future values of Y, beyond what can be predicted based on the past values of Y alone.

In our context, this may be phrased in the following terms:
Granger Causality: *A collection of bloggers is said to influence Blogger B_i if their collective past content (blog posts) is predictive of the future content of Blogger B_i, more so than the past content of Blogger B_i alone.*

Let $B_1 \dots B_G$ denote a community of G bloggers. To develop the above intuition further, we need to represent "content" and define "predictive". With each blogger, we associate *content variables*, which consist of frequencies of words relevant to a topic across time. Specifically, given a dictionary of K words and the time-stamp of each blog post, we record $w_i^{k,t}$, the frequency of the kth word for blogger B_i at time t [4]. Then, the *content* of blogger B_i at time t can be represented as $\mathbf{B}_i^t = [w_i^{1,t}, \dots, w_i^{K,t}]$. The input to our model is a collection of multiple time series, one for each blogger B_i: $\{\mathbf{B}_i^t\}_{t=1}^T$, where T is the timespan of our analysis. The output of our model is a causal graph that encodes causal relationships between bloggers. This is pictorially shown below.



(a) Input

(b) Output

The causal graph is constructed by processing each blogger one by one. At step j, we consider blogger B_j and pose the problem of predicting her content variables at time t, $\mathbf{B}_i^t$, in terms of the past content of the population, over a time window which is a parameter of our model. This is a multivariate regression with K-dimensional response vector $\mathbf{B}_j^t$ in terms of groups of variables $\{\mathbf{B}_1^{t-l}\}_{l=1}^d, \{\mathbf{B}_2^{t-l}\}_{l=1}^d, \dots, \{\mathbf{B}_G^{t-l}\}_{l=1}^d$, as illustrated in Figure 1.

In this part of the project, we will experiment with a new algorithm (a multivariate version of Group Orthogonal Matching Pursuit with a novel application to causal modeling) that selects certain variable groups (bloggers) as relevant in the regression, based on their ability to explain the response vector, namely the future content of blogger B_j. The selected groups then identify the causal links incident on B_j in the causal graph. Moreover, regression coefficients on each edge can be used to identify the most influential and the most influenced words.

Once the causal graph is constructed, one can define a family of influence measures on it that we call *GrangerRank*. The outdegrees of nodes in the causal graph, or the PageRank are examples of influence measures.

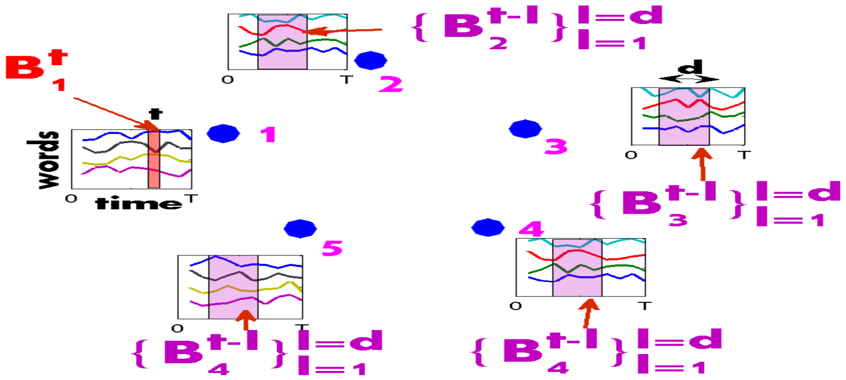
More technical details are available in the report [15].

4.1 Measuring Value of Causal Graphs

One can derive other measures of influence in twitter. For example, the number of followers is a natural measure. A proxy for a "ground truth" measure of influence of a microblogger is the probability of getting virally distributed (i.e., retweeted) in the near future. One way to benchmark the value of causal graphs in capturing influence more accurately than say the number of followers that a twitter account has, is to see if the causal influence ranking is more predictive of future

[4] this can also be computed as $\mathbf{X}'\mathbf{U}$ where $\mathbf{U}_{it} = 1$ if document i has timestamp t and 0 otherwise

Figure 1: Finding edges in the causal graph incident on Blogger 1. Shown is the response vector, and variable groups for multivariate regression.



B^t_1
t
words
0
time
T
1
2
3
4
5
d
$\{B^{t-l}_2\}^{l=d}_{l=1}$
$\{B^{t-l}_3\}^{l=d}_{l=1}$
$\{B^{t-l}_4\}^{l=d}_{l=1}$
$\{B^{t-l}_4\}^{l=d}_{l=1}$

retweeting. This can be done by measuring the rank correlation between future re-tweet frequency and the ranking returned by the causal model (i.e., via out-degrees or pagerank). Another natural baseline to compare against the historical retweet rate itself.

5 Combining Models: Understanding Topical Influence

Each microblogger can be associated with multiple time series representing usage of topics instead of words cross time. The final part of the project will attempt to combine the topic models with the causal models to arrive at a robust topical notion of influence.

6 Project Plan

Team A: Learning Topics with Sparse Low-rank Non-negative Matrix Factorizations

1. Explore the effect of sparsity on the tradeoff between the quality of topic models and scalability obtained due to reduction in memory requirements.
2. Qualitatively explore the performance of sparse NMF topic models on twitter datasets.
3. Compare competing topic modeling methodologies such as LDA against sparse low-rank non-negative factorizations as in [3].
4. Combine the topic model with causal models for influence working with Team B.

Team B: Inferring Key Influencers with Granger Causality

1. Implement GrangerRanks (Matlab)
2. Explore GrangerRanks on twitter datasets.
3. Benchmark them against measures like future retweet rates.
4. Build models at the level of topics working with Team A.

Note: This plan gives a sketch of some of the technical ideas we are interested in exploring. The project will allow for significant flexibility in exploring related technical directions based on evolving interests of team members and mentors.

References

[1] M. W. Berry, M. Browne, A.N. Langville, V. Paul Pauca, and R.J. Plemmons. Algorithms and applications of non-negative matrix factorizations.

[2] D. Blei and M.Jordan. Latent dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[3] J. Chang, J. Boyd Graber, S. Gerrish, C. Wang, and D.M. Blei. Reading tea leaves: How humans interpret topic models. In *NIPS*, 2009.

[4] A. Cichocki, R. Zdunek, A. H. Phan, and S. Amari. *Non-negative and Tensor Factorizations: Applications to Exploratory Multiway Data Analysis and Blind Source Separation*. Wiley, 2009.

[5] C. Ding, X. He, and H. D. Simon. On the equivalence of non-negative matrix factorizations and spectral clustering. *SDM*, 2005.

[6] C. Ding, T. Li, and W. Peng. On the equivalence between non-negative matrix factorizations and probabilistic latent semantic analysis. *Computational Statistics and Data Analysis*, 2008.

[7] N. Gilles and F. Glineur. Nonenegative factorization and the maximum edge biclique problem. In *CORE Discussion Paper*, 2008.

[8] C. Granger. Testing for causality: A personal viewpoint. *Journal of Economic Dynamics and Control*, 2:329–352, 1980.

[9] N.D. Ho. *Non-negative Matrix Factorizations: Algorithms and Application*. PhD thesis, Catholic University of Louvain, Department of Mathematical Engineering, 2008.

[10] T. Hoffman. Probabilistic latent semantic analysis. In *UAI*, 1999.

[11] Jingu Kim and Haesun Park. Sparse non-negative matrix factorizations for clustering. In *Tech Report (Georgia Tech) GT-CSE-08-01*, 2008.

[12] D. Lee and H.S. Seung. Learning the parts of objects using non-negative matrix factorizations. *Nature*, 1999.

[13] T. Li and C. Ding. The relationships among various nonnegative matrix factorization methods for clustering. *ICDM*, 2006.

[14] C.J. Lin. Projected gradient methods for non-negative matrix factorization. In *Neural Computation*, 2007.

[15] A.C. Lozano and V. Sindhwani. Inferring key inuencers in online communities using multivariate grouped granger causality. *Tech Report*, 2010.

[16] C. Manning, P. Raghavan, and H. Schutze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[17] P. Melville, V. Sindhwani, and R. Lawrence. Social media analytics: Channeling the power of the blogosphere for marketing insight. *Workshop on Information in Networks*, 2009.

[18] M. Steyvers and Tom Griffiths. *Probabilistic Topic Models*, chapter Latent Semantic Analysis: A Road to Meaning. Laurence Erlbaum, 2007.

[19] Stephen A. Vavasis. On the complexity of non-negative matrix factorizations. In *arXiv: 0708.4149*, 2007.

[20] Clustering Large Graphs via the SVD. P. drineas and a. frieze and r. kannan. *Machine Learning*, 2004.

[21] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorizations. *SIGIR*, 2003.