2017 International Conference on Data Science and Advanced Analytics

Learning Low-Rank Document Embeddings with Weighted Nuclear Norm Regularization

Lukas Pfahler*, Katharina Morik*, Frederik Elwert†, Samira Tabti† and Volkhard Krech†
*TU Dortmund University
44227 Dortmund, Germany
Email: firstname.lastname@tu-dortmund.de
†Ruhr-Universität Bochum
44801 Bochum, Germany
Email: firstname.lastname@rub.de

***Abstract*—Recently, neural embeddings of documents have shown success in various language processing tasks. These low-dimensional and dense feature vectors of text documents capture semantic similarities better than traditional methods. However, the underlying optimization problem is non-convex and usually solved using stochastic gradient descent. Hence solutions are most-likely sub-optimal and not reproducible, as they are the result of a randomized algorithm.**

We present an alternative formulation for learning low-rank representations.Instead of explicitly learning low-dimensional features, we compute a low-rank representation implicitly by regularizing full-dimensional solutions. Our approach uses the weighted nuclear norm, a regularizer that penalizes singular values of matrices. We optimize the regularized objective using accelerated proximal gradient descent. We apply the approach to learn embeddings of documents.

We show that our approach outperforms traditional convex approaches in a numerical study. Furthermore we demonstrate that the embeddings are useful for detecting similarities on a standard dataset. Then we apply our approach in an interdisciplinary research project to detect topics in religious online discussions. The topic descriptions obtained from a clustering of embeddings are coherent and insightful. In comparison to existing approaches, they are also reproducible.

An earlier version of this work stated, that the weighted nuclear norm is a convex regularizer. This is wrong – the weighted nuclear norm is non-convex, even though the name falsely suggests that it is a matrix norm.

I. Introduction

High-dimensional data is everywhere, but processing and analyzing high-dimensional data remains challenging. Low rank models are used to uncover structure in high dimensional data. They assume that the high-dimensional matrix of observations is the, possibly noisy, image of a latent, low-rank matrix under a model-function. We try to recover this latent representation given the data matrix, thus obtaining a low-dimensional representation of the data. Possible applications are manifold, examples include text classification [1], information retrieval [2], collaborative filtering [3], image denoising [4] and pattern mining [5]. More broadly speaking, low rank applications are used as a preprocessing step for machine learning or data mining tasks in techniques like feature extraction or dimensionality reduction [6].

Low rank modeling is particularly useful in social sciences or digital humanities. Whether we work with records of social interactions in social networks or human-generated contents like tweets or direct messages, data is often not-only high-dimensional, but also complex, unstructured and noisy. We can use low rank models to uncover the latent structures or underlying factors. For instance, topic models identify word fields in text collections [7]; applications in social network analysis include community detection [8] and link prediction [9]. Recently, embedding learning has become a popular approach to the analysis of high-dimensional, sparse structures. We want to obtain a low-rank representation of, for instance, words in contexts [10], vertices in graphs [11] or documents in document collections [1]. These low-dimensional and dense embeddings are trained to represent their high-dimensional counterparts with simple prediction models [12], but exploit regularities in the data by assuming a low-rank structure. They are especially useful for capturing similarities between data points.

Particularly in scientific domains, reproducibility is crucial. Many approaches for training low-rank models formulate optimization problems with explicitly factorized models $X = UV^T$ and optimize over both factors U, V simultaneously. Objectives like this are non-convex and have many local optima. Optimization routines apply randomized techniques like stochastic gradient descent [13] and return local optima, thus results differ between multiple runs. This heuristic nature is often met with skepticism, especially researchers in the humanities question the significance of the findings obtained with these algorithms. This motivates the search for methods that find globally optimal solutions and shows the importance of understanding the structure of the space of possible solutions.

We demonstrate its usefulness with a particular model for learning low-rank embeddings of text documents that can be trained efficiently by using a hierarchical softmax tree to model word probabilities [14]. We show that this model can be used for topic modeling in a digital humanities application. In collaboration with scientists in religious studies we identify the topics discussed in religious online discussion boards.

The rest of this paper is structured as follows: In Section II we introduce an optimization framework for low rank models.We use the weighted nuclear norm and propose to use a simple weighting scheme with intriguing properties.

978-1-5090-5004-8/17 $31.00 © 2017 IEEE
DOI 10.1109/DSAA.2017.46

In Section III we propose a new method to obtain low rank embeddings of text documents based on document embeddings [1]. After presenting related work in Section IV, we evaluate our approach empirically in Section V. This evaluation is based on quantitative analysis on a standard dataset as well as qualitative analysis in a social science setting. We conclude this paper in Section VI and provide an outlook to future research questions.

II. Training of Low Rank Models

In machine learning or data mining applications, we often model data as a function of a low-rank matrix and fit the model to the data by minimizing a loss function. Generally, we are interested in solving optimization problems of the form

$$\min f(X) \text{ s.t. } \text{rank}(X) \leq k \tag{1}$$

with low rank solutions. Following the motivation above, $f(X)$ is the composition of a model function and a loss function. It measures how well the model fits the low-rank representation X to the given data. We limit our analysis to convex and smooth functions $f(X)$. In this section, we first formally define low rank matrices, then we introduce the weighted nuclear norm, a regularizer that favors low rank solutions, and finally we outline the accelerated proximal gradient descent method, which we use to optimize our objective.

A. Low Rank Matrices and Optimization

Let $\Omega_k := \{X \mid X \in \mathbb{R}^{m\times n}, \text{rank}(X) \leq k\}$ denote the set of matrices with rank at most k. Unfortunately Ω_k is not convex, as for $\lambda \in [0,1], X, Y \in \Omega_k$ with rank(X) =rank$(Y) = k$ it holds that rank$(\lambda X + (1-\lambda)Y)$ can be as large as $2k$ and thus $(\lambda X + (1-\lambda)Y)$ is not necessarily in Ω_k. Consequently steepest descent optimization methods will likely get stuck in local minima.

Minimizing a function $f(X)$ over the domain Ω_k can be written as an unconstrained optimization problem

$$\min_{X\in\mathbb{R}^{m\times n}} f(X) + \delta(X) \text{ with } \delta(X) = \begin{cases} 0 & \text{if } X \in \Omega_k \\ \infty & \text{otherwise.} \end{cases} \tag{2}$$

Obviously $\delta(X)$ is a non-convex function since Ω_k is a non-convex set.

An important operator in constrained optimization is the proximity operator

$$\text{prox}_g(X) = \arg\min_Y \frac{1}{2}||X-Y||_F^2 + g(Y). \tag{3}$$

For δ the proximity operator projects any matrix to the nearest matrix with respect to the Frobenius norm that has rank at most k. We apply this operator in numerical optimization routines to ensure that the solution we computed still is a member of Ω_k, for instance after taking a step in the direction of the gradient. Let $X = U\Sigma V^T$ be a singular value decomposition with $\Sigma = \text{diag}(\sigma)$, $\sigma_1 \geq \sigma_2 \geq ... \geq 0$. Then a closed form solution is

$$\text{prox}_\delta(X) = \arg\min_Y \frac{1}{2}||X-Y||_F^2 + \delta(Y) = U\text{diag}(\bar{\sigma})V^T \tag{4}$$

with $\bar{\sigma} = [\sigma_1, ..., \sigma_k, 0, ..., 0]^T$. This operation is also called truncated singular value decomposition for a fixed rank k.

We propose to replace $\delta(X)$ with a regularizer, the weighted nuclear norm [15], [16]. The regularizer penalizes non-zero singular values, favoring solutions with low rank, as the rank of a matrix is the number of its non-zero singular values. In this paper, we propose the following weighting scheme and define the weighted nuclear norm as follows: For small $1 > \varepsilon > 0$ and $X = U\text{diag}(\sigma)V^T$ with non-increasing σ let

$$||X||_\varepsilon = \sum_{i=1}^{k} \varepsilon \cdot \sigma_i + \sum_{i=k+1}^{\min m,n} \frac{1}{\varepsilon} \cdot \sigma_i =: \langle \theta, \sigma \rangle, \tag{5}$$

where we introduce the weight vector θ for notational convenience. Like $\delta(X)$, it has a proximal operator with closed form solution called singular value thresholding [15]. For a singular value decomposition $X = U\Sigma V^T$ it returns

$$\text{prox}_\varepsilon(X) = U\text{diag}(\sigma - \theta)_+ V^T \tag{6}$$

where

$$(\sigma - \theta)_+ = [\max(0, \sigma_1 - \theta_1), ..., \max(0, \sigma_l - \theta_l)] \tag{7}$$

with $l = \min(m,n)$ decrements the singular values by θ and thresholds all negative values to 0.

Proof. Let $X = U\Sigma V^T$ and $\hat{X} = U\text{diag}(\sigma - \theta)_+ V^T$. It suffices to show that

$$(X - \hat{X}) \in \partial ||\hat{X}||_\varepsilon$$

Since σ is non-increasing and θ is non-decreasing, there is i_0 such that $\sigma_i - \theta_i > 0$ if and only if $i \leq i_0$. We rewrite $X = U_{(1)}\Sigma_{(1)}V_{(1)}^T + U_{(2)}\Sigma_{(2)}V_{(2)}^T$ where $U_{(1)} \in \mathbb{R}^{m\times i_0}, \Sigma_{(1)} \in \mathbb{R}^{i_0\times i_0}, V_{(1)} \in \mathbb{R}^{n\times i_0}$ contain the first i_0 columns of U, Σ, V respectively and $U_{(2)}, \Sigma_{(2)}, V_{(2)}$ contain the remaining entries. Analogously $\theta_{(1)}$ contains the first i_0 entries of θ. Consequently, we have

$$(X - \hat{X}) = U_{(1)}\text{diag}(\theta_{(1)})V_{(1)}^T + U_{(2)}\Sigma_{(2)}V_{(2)}^T$$

Following Watson [17], we have to show that the diagonal entries are a subgradient of the gauge function $\phi(\sigma) = \langle \theta, \sigma \rangle, \sigma \geq 0$:

$$\begin{pmatrix} \theta_{(1)} \\ \sigma_{(2)} \end{pmatrix} \in \partial\phi \begin{pmatrix} \sigma_{(1)} - \theta_{(1)} \\ 0 \end{pmatrix}$$

The subgradient is given by $\partial\phi(z) = \{x \mid x_i \leq \theta_i \text{ and } z_i > 0 \Rightarrow x_i = \theta_i\}$. Since $\sigma_{(1)} - \theta_{(1)} > 0$ and $\sigma_{(2)} < \theta_{(2)}$ the diagonal entries are a subgradient, which concludes the proof. □

Interestingly, as ε approaches 0, $\text{prox}_\varepsilon(X)$ converges pointwise to $\text{prox}_\delta(X)$:

$$\forall X \lim_{\varepsilon\to 0} \text{prox}_\varepsilon(X) = \text{prox}_\delta(X). \tag{8}$$

B. Accelerated Proximal Gradient Descent

To solve an optimization problem of the form

$$\arg \min_{X \in \mathbb{R}^{m \times n}} \underbrace{f(X) + ||X||_\varepsilon}_{:=F(X)} \tag{9}$$

where $f(X)$ is a convex and smooth function, we can apply proximal gradient descent [18].

With a stepsize $\alpha > 0$ and starting at an initial solution $X^{(0)}$, we iterate the following steps for $t = 1, ...$

$$X^{(t+\frac{1}{2})} = X^{(t)} - \alpha \nabla f(X^{(t)}) \tag{10}$$

$$X^{(t+1)} = \text{prox}_{\alpha,\varepsilon}(X^{(t+\frac{1}{2})}) \tag{11}$$

The proximal operator for stepsize α is defined as

$$\text{prox}_{\alpha,\varepsilon}(Y) = \arg\min_X \frac{1}{2\alpha}||X - Y||_F^2 + ||X||_\varepsilon \tag{12}$$

$$= \arg\min_X \frac{1}{2}||X - Y||_F^2 + \langle \alpha\theta, \sigma(X) \rangle \tag{13}$$

thus we can still apply singular thresholding as in (6) with modified weights $\alpha\theta$.

It has been shown empirically that the convergence rate can be improved by applying Nesterov acceleration [19]. Starting with an initial solution $X^{(0)}$ and $s^{(0)} = 1$ we iterate the following steps for $t = 1, ...$

$$s^{(t)} = \frac{1 + \sqrt{1 + (2s^{(t-1)})^2}}{2} \tag{14}$$

$$Y^{(t)} = X^{(t)} + \left(\frac{s^{(t-1)} - 1}{s^{(t)}} \right) (X^{(t)} - X^{(t-1)}) \tag{15}$$

$$X^{(t+\frac{1}{2})} = Y^{(t)} - \alpha \nabla f(Y^{(t)}) \tag{16}$$

$$X^{(t+1)} = \text{prox}_{\alpha,\varepsilon}(X^{(t+\frac{1}{2})}) \tag{17}$$

This modified algorithm is called accelerated proximal gradient descent [18].

Note that the solution of $\text{prox}_\varepsilon(X)$ can have rank larger than k. Thus regularization with the weighted nuclear norm allows the optimization routine to choose a higher rank if the data requires it.

III. Regularized Training of Document Embeddings

Low rank representations have a long tradition in natural language processing. We need to represent texts, sequences of discrete symbols, in fixed-dimensional vector spaces for classification, clustering or similarity detection. When we use the bag-of-words representation that expresses each word with a component in a vector space [20], text documents are high-dimensional. A collection of n documents with a vocabulary of size m is expressed as a term-document matrix $F \in \mathbb{R}^{m \times n}$ where $f_{w,d}$ indicates how often the word w appeared in document d. Using this sparse representation makes is difficult to detect similarities between two documents, as two texts on the same topic might not share a single word. Principal component analysis (PCA) or singular value decomposition (SVD), also called latent semantic indexing (LSI) [2] in the context of information retrieval, identify latent factors of the term-document matrix $f_{w,d}$. We use only the first k of these factors, thus obtaining a low-rank representation of documents. The general idea is to exploit the fact, that certain words often occur together, hypothesizing that they have a similar semantic.

Le and Mikolov recently presented document embeddings [1], an extension of the popular word embedding algorithm word2vec [10]. We use their approach as a starting point to develop a document embedding algorithm based on optimization and regularization.

A. Document Embeddings

The key idea of document embeddings is to encode the high-dimensional and sparse distribution of words in a given document in a low-dimensional, dense vector. To this end, we introduce real-valued embeddings for words, denoted by $u_w \in \mathbb{R}^k$, and documents, denoted by $v_d \in \mathbb{R}^k$ with $k \ll m$. We model the probabilities of a word given a document by

$$P(w|d) = \exp\left[\langle u_w, v_d \rangle - Z(d)\right] \tag{18}$$

where the partition function Z is defined as

$$Z(d) = \log \sum_{w'=1}^{m} \exp \langle u_{w'}, v_d \rangle \tag{19}$$

We train document embeddings given n documents by maximizing the likelihood of the model. To this end we minimize negative log-likelihood

$$\arg\min_{U,V} \sum_{d=1}^{n} \sum_{w=1}^{m} -f_{w,d} \log P(w|d) \tag{20}$$

This objective is non-convex and thus prone to local minima. Le and Mikolov propose to train embeddings by minimizing the objective using stochastic gradient descent (SGD). Since SGD is a randomized algorithm, the results vary between different runs of the training algorithm.

B. Hierarchical Softmax

Computing the partition function Z for large vocabulary sizes m is computationally expensive. A common practice in language modeling and word embedding learning is to use the so-called hierarchical softmax model [14], [10] to obtain word probabilities given an embedding. The probabilities $P(w|d)$ are factored according to a binary tree with m leaf nodes corresponding to the words. Each of the $(m-1)$ inner nodes corresponds to a binary random variable that indicates whether a path from root to leaf chooses the left or right child. Thus each word corresponds to a path from the root of the tree to the corresponding leaf node. The probability of a words is defined as the product of all probabilities along the path. Formally we define the probabilities at inner nodes v, $P_v(\text{left}|d) = 1 - P_v(\text{right}|d)$, a function that gives all inner nodes from root to leaf for a word path(w) and an indicator function $\psi(v, w) \in \{\text{left}, \text{right}\}$ that indicates if the path from

the root to w turns left or right at the inner node v. Now the word probability can be written as

$$P(w|d) = \prod_{v \in \text{path}(w)} P_v(\psi(v,w)|d). \tag{21}$$

Le and Mikolov model the probabilities at the inner nodes using sigmoid functions and inner products of node embeddings u_v and document embeddings v_d such that

$$P_v(\text{left}|d) = \sigma(\langle u_v, v_d \rangle) = \frac{1}{1 + e^{-\langle u_v, v_d \rangle}}. \tag{22}$$

We propose to substitute the scalar product, the source of non-convexity in the original formulation, for an entry in a low rank matrix $X \in \mathbb{R}^{(m-1)\times n}$. Because X is low-rank, every entry $X_{vd} = \langle u_v, v_d \rangle$ for a decomposition $X = UV^T$. We define

$$P_v(\text{left}|d) = \sigma(\langle u_v, v_d \rangle) = \frac{1}{1 + e^{-X_{vd}}}. \tag{23}$$

To train the embeddings, we minimize negative log-likelihood. For a single pair of document and word we obtain

$$-\log P(w|d) = \sum_{v \in \text{path}(w)} \log(1 + e^{-\xi(v,w)\cdot X_{vd}}) \tag{24}$$

where we define ξ for notational convenience:

$$\xi(u,v) = \begin{cases} 1 & \text{if } \psi(u,v) = \text{left} \\ -1 & \text{otherwise.} \end{cases} \tag{25}$$

We see that minimizing negative log-likelihood is a convex and smooth function of X. The full unregularized training objective is

$$f(X) = \sum_{d=1}^{n} \sum_{w=1}^{m} f_{w,d} \sum_{v \in \text{path}(w)} \log(1 + e^{-\xi(v,w)\cdot X_{vd}}) \tag{26}$$

which is also convex. We propose to minimize the regularized objective $F(X) = f(X) + ||X||_\varepsilon$ using accelerated proximal gradient descent as presented above to obtain a low-rank solution. The gradient of $f(X)$ is given by

$$\frac{\partial f(X)}{\partial X_{vd}} = f_{v,d,\text{left}} \cdot \sigma(X_{vd}) + f_{v,d,\text{right}} \cdot (\sigma(X_{vd}) - 1) \tag{27}$$

where $f_{v,d,\text{left}}$ indicates how many words in d use a path through v and choose the left child of v, $f_{v,d,\text{right}}$ is defined analogously. Also we introduce $f_{v,d}$ for the sum of both.

Finally we show that $f(X)$ has a Lipschitz-continuous gradient; we will use the Lipschitz constant to choose the step size. The sigmoid function is Lipschitz-continuous with $|\sigma(x) - \sigma(y)| \leq \frac{1}{4}|x - y|$. Thus we can show that $\nabla f(X)$ is Lipschitz-continuous

$$||\nabla f(X) - \nabla f(Y)||_F \tag{28}$$

$$= \left[\sum_{d=1}^{n} \sum_{v=1}^{m-1} f_{v,d}^2 (\sigma(X_{vd}) - \sigma(Y_{vd})^2 \right]^{\frac{1}{2}} \tag{29}$$

$$\leq \frac{1}{4} \max_d |d| \left[\sum_{d=1}^{n} \sum_{v=1}^{m-1} (X_{vd} - Y_{vd})^2 \right]^{\frac{1}{2}} \tag{30}$$

$$= \frac{1}{4} \max_d |d| \cdot ||X - Y||_F \tag{31}$$

with Lipschitz-constant $L := \frac{1}{4} \max_d |d|$. Hence we choose the stepsize $\alpha = \frac{4}{\max_d |d|}$.

We obtain the document embeddings by taking the final iterate X, decomposing it as $X = U\Sigma V^T$ and returning $\sqrt{\Sigma} V^T$.

C. Implementation Details

We factor the word probabilities according to a Huffman tree that is built using the word frequencies in the training data. This way the average length of paths from root to leaf is minimal. Also the number of non-zero elements in the gradient is minimal. We can exploit this sparsity to reduce the computation time required for the singular value decomposition that is necessary to compute the proximal operator. We can compute SVD($X^{(t+\frac{1}{2})}$) using the Lanczos bidiagonalization with partial reorthogonalization method for computing the truncated singular value decomposition [21]. This method can compute the truncated singular decomposition of A efficiently when the matrix-vector products Ax and $A^T y$ can be computed efficiently [3]. In our case, $X^{(t+\frac{1}{2})}$ is the sum of a low-rank matrix $Y^{(t)}$ and a sparse matrix $\nabla f(Y^{(t)})$. The matrix-vector-products with $Y^{(t)}$ can be computed in $\mathcal{O}(k(n+m))$ if k is the rank of $Y^{(t)}$. For $\nabla f(Y^{(t)})$ we can compute the product in $\mathcal{O}(s)$ where s is the number of non-zero elements of the gradient. If we assume an average length of paths in the softmax tree of $\log m$, we can compute the matrix-vector-product in $\mathcal{O}(\tau \log m)$, where τ is the number of tokens in the dataset. So in total the product has runtime in $\mathcal{O}((n+m)k + \tau \log m)$ which is substantially cheaper than the computational cost of the naive matrix-vector product $\mathcal{O}(m \cdot n)$, as long as $\tau \ll m \cdot n$, a reasonable assumption for natural language documents.

It suffices to compute a truncated SVD, because the rank of the iterates $X^{(t)}$ does not vary drastically between iterates, an intuition that is confirmed by pertubation bounds for singular values. Hence we can compute a truncated SVD with rank $(1+c)k^{(t-1)}$ where $c > 0$ is a small constant and $k^{(t-1)}$ is the rank of the last iterate. If the last singular value of the resulting truncated SVD would be non-zero after applying the proximal operator, we increase c and recompute the truncated SVD until the specified rank is sufficiently large. This way we can compute the exact proximal operator without computing a full SVD.

We initialize $X^{(0)} = 0$. In combination with the Huffman tree, this yields initial word probabilities approximately equal to the prior word probabilities estimated using the document collection.

We provide an open source C++ implementation of our algorithm at https://bitbucket.org/Whadup/doc2vex. It uses PROPACK[1], a Fortran implementation of Larsen's SVD algorithm [21].

IV. Related Work

Optimization methods for learning low rank models have recently gained a lot of attention. We distinguish convex

[1] sun.stanford.edu/~rmunk/PROPACK/

approaches, non-convex approaches and manifold approaches. Convex approaches compute low rank solutions by applying regularization to full-rank matrices. Traditionally we use the squared Frobenius norm, the sum of squared singular values, because it is differentiable. It favors solutions with small singular values and we can write the Frobenius norm as the L2-norm of the vector of singular values. Analogously the nuclear norm $||X||_* = \sum_i \sigma_i$ is defined as the sum of the singular values, thus we can write it as $||X||_* = ||\sigma||_1$. As the L1-norm induces sparsity, the nuclear norm induces low rank [22]. Furthermore, we can show that $||X|| \cdot ||X||_2^{-1}$ is the convex envelope of the rank, i.e. the largest convex function that lower bounds rank(X). The nuclear norm can serve as an approximation to the non-convex rank-function because it is easier to minimize [22].

The weighted nuclear norm [15] used in this work applies a non-decreasing weight vector to penalize each singular value individually. Thus it is related to the sorted-L1 norm for vectors [23]. It gives us fine-grained control over the trade-off between low-dimensional features and a good loss value $f(X)$. When we want to extract low-dimensional, dense features from high dimensional data, we expect that a certain number of features is needed in order to capture the characteristics of the input data. Hence we can tolerate a certain amount of non-zero singular values and only wish to penalize further ranks. Thus applying different weights for different weights with weighted nuclear norm is useful.

Haeffele et al. propose the so-called projective tensor norm, which constructs complex, problem-specific regularizers for matrices by combining vector norms that are evaluated on the columns of a factored matrix $X = AZ^T$. The authors show that, while the problem is no longer convex because of the explicit factorization, under certain conditions local optima are global [4].

There are many non-convex methods based on explicit factorizations. Recent research investigates the conditions under which the local minimizers obtained using this methods are global minimizers [24], [25], [26]. For instance Ge et al. show that for matrix completion problems with symmetric target matrices and $X = ZZ^T$, the non-convex optimization over Z has no bad local optimums and that the global optimum can be found with gradient descent [25]. Sun and Luo proof a similar result for general matrices, although they require an expensive initialization procedure to generate a start solution near the global optimum [24]. These methods all use a loss function that sums the squared errors of observed matrix entries and their respective low-rank approximation. We are curios if similar results can be achieved for the negative log-likelihood function used in this work.

Another non-convex method, that does not use explicit factorizations is projection gradient descent. We use gradient descent and $X^{(t+1)} = \text{prox}_\delta(X^{(t)} - \nabla f(X^{(t)}))$ with prox_δ as defined in (4). Thus after each gradient step, we map the solution back to the set of matrices with rank at most k by truncated SVD. We have already seen that using the weighted nuclear norm with the weighting scheme proposed in this work are closely related via (8). Future work should follow in this direction and investigate the characteristics of local minimizers obtained by projected gradient descent as well as factored models like the original formulation of document embeddings [1]. It remains unclear if all local minimizers are good local minimizers or even global minimizers.

An interesting alternative to optimization approaches that operate in real-valued spaces are optimization methods for manifolds. The set of low-rank matrices is an embedded manifold in the set of real-valued matrices [27], [28]. Instead of using the natural gradient, we use the Riemannian gradient, which is the gradient mapped to the tangent space of the manifold. After taking a step in the direction of the Riemmanian gradient, we map the solution back to the manifold using so-called retractions [27]. The most-popular retraction is the proximal operator $\text{prox}_\delta(X)$ that requires computing the SVD of the solution. Shalit et al. propose to use a different retraction that is cheaper to compute when the rank of the gradient is small [29]. This is the case when we use stochastic gradients that are defined for single documents.

Finally this work is related to other embedding learning algorithms for text documents. Following the success of word embeddings [10], various approaches to represent text as a combination of word vectors have been proposed, ranging from simple addition to more sophisticated neural network structures [30]. These methods heavily rely on pre-computed word embeddings, which may introduce bias that exists in the underlying text collection used for training word embeddings [31]. This is potentially troubling for scientific applications, but particularly for social science studies. The document embedding method [1] as well as the method presented in this work uses the same text collection to derive word embeddings and document embeddings. However, the non-convex formulation of Le and Mikolov allows the user to initialize the word embeddings with pre-computed vectors and then finds a locally optimal solution starting from these pre-trained vectors.

V. Experimental Evaluation

We first evaluate the trade-off of weighted nuclear norm regularization between low-rank solutions and good approximation of the training data in a numerical study. Then we investigate our hypothesis that document embeddings capture similarities in documents using a task where ground-truth labels are available. We compare the usefulness of different text representations for clustering the 20newsgroup dataset with k-means. Then we apply our approach in a digital humanities application and detect topics in online discussion boards using hierarchical clustering, a deterministic clustering algorithm.

A. Optimization

We start evaluating the proposed algorithms on a small synthetic dataset. The following evaluation is guided by the hypothesis that the weighted nuclear norm yields better trade-offs between low rank and loss than other convex regularizers.

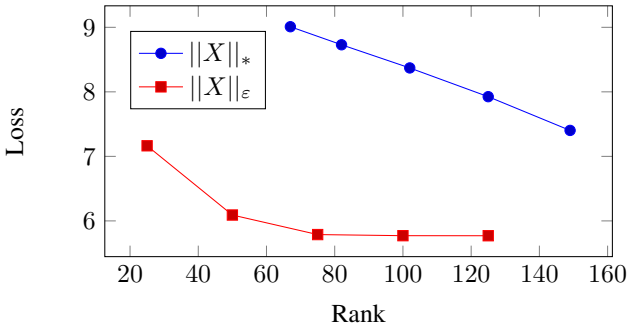

Fig. 1. Loss vs. Rank for different regularizers. We compare the nuclear norm and the weighted nuclear norm proposed in this paper. We see that the weighted nuclear norm yields a better trade-off between loss and rank.

We generate 500 documents with a vocabulary size of 1000 distinct tokens by randomly creating 25 categorical distributions, sampling one distribution per document, sampling a document length uniformly between 10 and 100 and repeatedly sampling tokens from the selected categorical distribution.

Then we compare the weighted nuclear norm regularization with the nuclear norm for different target ranks k and regularization constants. For the weighted nuclear norm, we set $\varepsilon = \frac{1}{50}$ which is sufficiently large to obtain solutions with rank close to the target rank $k \in \{25, 50, 75, 100, 125\}$. We run the optimization until convergence and report the rank and loss of the solution.

As we can see in Figure 1, the constant weights of the nuclear norm yield the worst solutions. In comparison, using the weighted nuclear norm yields significantly better loss values. This fits our intuition that we should only penalize the singular values corresponding to high ranks. Penalizing the first singular values interferes with fitting the model.

The training is also faster when the rank is lower, because the runtime of the truncated SVD computation depends quadratically on the rank. Thus, in addition to the better fitting-capabilities of the weighted nuclear norm, training time is also faster.

B. Text Clustering

In this first experiment, we evaluate how useful the representations are for detecting similarities between documents. To this end we use the clustering algorithm k-means that requires good similarity measures to find good clusterings. We can evaluate the quality of the clustering using ground-truth labels.

We use the popular 20-newsgroup dataset[2], which consists of approximately 20k emails from 20 newsgroup categories. We remove all e-mail headers and quotes and prune very short and very long documents, finally obtaining 18,470 documents. Furthermore, we prune the vocabulary to contain only words that occur 10 times or more, resulting in a vocabulary of $20,819$ distinct words.

We train a representation with rank $k = 300$ using our approach. We set $\epsilon = \frac{1}{500}$ and run 1,000 iterations of the accelerated proximal gradient descent algorithm. This way we obtain embeddings that achieve a perplexity per word of 174.49. We compare against the original doc2vex [1] and LSI, which we use to train representations with 300 dimensions. For this comparison, we rely on the implementations available in the gensim package[3]. For doc2vec, we run 200 training epochs with a fixed stepsize of 0.01. It uses the exact same preprocessing and the same Huffman tree as our approach. The embeddings trained this way achieve a perplexity of 184.43. Furthermore we compare against the bag-of-words representation and the TF-iDF representation, which re-weights the bag-of-words [32].

TABLE I
CLUSTER PERFORMANCE

Approach	Accuracy	Purity $k = 20$	Purity $k = 50$
BOW	$20.00\% \pm 1.20\%$	$21.49\% \pm 1.08\%$	$26.42\% \pm 0.92\%$
TF-iDF	$42.10\% \pm 3.20\%$	$43.40\% \pm 3.11\%$	$50.67\% \pm 2.32\%$
LSI	$13.40\% \pm 1.09\%$	$14.64\% \pm 0.91\%$	$19.69\% \pm 1.09\%$
doc2vec	$\mathbf{54.60\% \pm 1.90\%}$	$\mathbf{55.64\% \pm 1.73\%}$	$\mathbf{58.67\% \pm 0.84\%}$
this work	$52.50\% \pm 1.70\%$	$53.42\% \pm 1.48\%$	$54.13\% \pm 0.95\%$

We use the k-means algorithm to cluster the document representations into 20 clusters. We use cosine similarity as a distance measure. To improve stability of the results, we run k-means 10 times and use the clustering with the highest intra-cluster similarity. We compare the clustering with the ground-truth categories and report accuracy and purity. For accuracy, we compute the best bijection between clusters and labels and report classification accuracy. For purity we assign each cluster its most frequent class label and report classification accuracy. No unsupervised clustering metrics, for instance average intra-cluster distances, were reported, because they are meaningless when we switch representations.

In Table I, we see that the document embeddings significantly outperform traditional methods for text representation. This suggest that embeddings indeed capture similarities. The doc2vec approach outperforms this work by a small margin. We hypothesize that the stochastic gradient descent optimization works as an implicit regularizer [33]. Future research should look into adding more explicit regularization to our convex model, for instance Frobenius Norm regularization $||X||_F$.

We omit a detailed comparison of running times, since our approach is orders of magnitude slower. This is mainly due to the singular value decompositions that have to be computed each iteration.

C. Topic Identification in Religious Online Discussion Boards

In an interdisciplinary research project in collaboration with scholars in the study of religion, we investigate, how religious authorities arise in online communities and how they compare to traditional religious authority figures. To this end we have crawled online discussion boards and compiled a text collection of online discussion threads. In the first phase of the project we want to identify the topics discussed in the

[2]http://qwone.com/~jason/20Newsgroups/

[3]https://radimrehurek.com/gensim/

TABLE II
EXAMPLES OF NEAREST NEIGHBOR PAIRS: WE SHOW THREAD-TITLES FOR EXAMPLE QUERIES AND THEIR RESPECTIVE NEAREST NEIGHBORS. (TRANSLATED FROM GERMAN)

Query	Result
Sex in marriage	Looking for Christians, who did NOT have sex before marriage
If we confess our sins, ...	What was the christening of Johannes?
Is abortion after rape really OK?	catholic media award: price money funding pro-abortion rally
Cooking in February	Cooking in May

text collection. The discussion board analyzed in this paper is German, all results presented here will be translated to English. Recently topic modeling [34] has gained a lot of usage in digital humanities research. Given a text collection it uses a probabilistic model to identify latent word distributions called topics and represent documents as mixtures of these topics. The topics are presented to social scientists for further analysis, most often in the form of lists containing the most probable words per topic. LDA applies randomized inference techniques like Gibbs sampling or variational inference, hence results differ between multiple runs of the algorithm.

We propose an alternative: First we compute low-rank embeddings with our approach. Then we use a deterministic clustering algorithm to cluster the embeddings. Finally we obtain topic descriptions from the clusters.

Before computing the embeddings, we apply standard preprocessing to the documents. We extract the discussion posts from the raw html crawls and remove all html markup, obtaining one text document for each thread. Then we split the text into tokens with a simple tokenizer based on a regular expression. We remove stop words and tokens longer than 15 characters or shorter than 3 characters. Then we prune the vocabulary to contain only words that occur more than 10 times, thus obtaining a vocabulary of size 57,621. The total number of documents is 20,178.

We compute embeddings with rank $k = 300$ and $\varepsilon = \frac{1}{500}$ by running 1,000 iterations of the accelerated proximal gradient descent algorithm. The embeddings have rank 308 and achieve a per word perplexity of $1,295$ and the improvement of the loss function in the last iteration has been less than 0.005%, thus the solution is sufficiently close to convergence. The embeddings obtained this way are useful to detect semantic similarities. We can find the similar documents by finding the nearest neighbor under cosine-similarity. This is especially useful for selecting the relevant documents for manual analysis. In Table II we show some examples of nearest neighbor queries. We only report the title of the discussions, but the semantic similarities of the discussions are apparent.

To automate the exploratory analysis of the text collection exploiting embedding similarities, we use a clustering algorithm that is deterministic: agglomerative hierarchical clustering. Unlike partitioning algorithms like k-means or db-scan it is does not utilize randomization. The k-Means algorithm greedily optimizes a non-convex objective starting with a random solution resulting in a local minimum that is not stable over multiple runs. Hierarchical clustering works by merging the most-similar clusters in a bottom-up fashion, starting with one cluster per document. We use the cosine similarity measure and use the complete-link strategy that merges the two clusters with largest minimum cosine-similarity between a pair of documents from each cluster.

In our application domain, hierarchical clustering has many advantages. We can adjust the number of clusters we wish to analyze by pruning the cluster hierarchy, as indicated in Figure 2. This can be done without recomputing embeddings and the cluster hierarchy. In contrast, computing a different number of clusters with LDA requires recomputing the whole model, which is often time-consuming. Also, the hierarchy allows us to interactively split and join clusters according to the uncovered tree structure. This interactive nature allows the user to find the right granularity of the clustering for their respective application. Other clustering algorithms require the user to guess hyperparameters like the number of clusters or the diameter of clusters to control granularity.

To obtain descriptions of the computed clusters, we rely on the more interpretable bag-of-words representation of the data. We compute the word frequency vectors for each node of the cluster in a bottom-up fashion. Each node is associated with a vector that contains aggregated word counts for all documents below the node. These cluster description can be viewed as topics. To highlight the differences between the clusters, it is useful to not report term-frequencies $f_{w,c}$ for each cluster c, but tf-idf scores. For each token we count the number of clusters on the selected level of the hierarchy that contain the token. By dividing the counts by the number of clusters we obtain the document frequency of a token d_t. Finally we report the tf-idf score $f_{w,c} \log \frac{1}{d_t}$ for each cluster and token.

For our analysis, we prune the cluster hierarchy to 100 leaf clusters. We provide a sample of 10 cluster descriptions in Table III, the first seven of which are coherent and insightful and the last three are less so. Using our approach we identified that discussions span from heavily religious topics, like differences between Christian denominations, to topics of everyday life like cooking. In between these extremes there are topics like family planning, relationships, death or issues more controversial in religious communities, like euthanasia, abortion and homosexuality. All these topics are debated using religious arguments as well as humanist, political, atheist or scientific arguments. To the social scientist, the documents in these clusters provide insight into the degree of pluralism within the online community and the general attitude towards issues. The threads identified can also help to analyze usergroups in the community. We hope to identify groups like conservatives or moderates by analyzing the threads on the level of individual posts.

We also identify topics that are incoherent or pointless. For instance, Topic 8 in Table III mostly consists of words like 'scroll', 'mark' or 'post' generated by the discussion board software. Indeed manual analysis of the documents shows that the cluster contains mostly threads that have

TABLE III
SAMPLE CLUSTER DESCRIPTIONS DISCOVERED USING DOCUMENT EMBEDDINGS, HIERARCHICAL CLUSTERING AND TF-IDF. SECOND LINE PROVIDES THE NUMBER OF DOCUMENTS IN EACH CLUSTER. (TRANSLATED FROM GERMAN)

Topic 1	**Topic 2**	**Topic 3**	**Topic 4**	**Topic 5**	**Topic 6**	**Topic 7**	**Topic 8**	**Topic 9**	**Topic 10**
# 343	# 143	# 89	# 115	# 218	# 325	# 206	# 480	# 100	# 124
tasty	pregnant	punishment	euthanasia	partner	birth	Eucharist	scroll	mercy	account
salad	pregnancy	sin	critic	sex	congratulations	ecumenism	mark	teacher	alcohol
recipe	doctor	hate	AFD	relationship	baby	Protestant	tree	israel	data
cook	pill	lust	death	marriage	warm-hearted	Catholic	2015	creation	facebook
vegetables	pain	act	die	boyfriend	excited	Luther	healing	landlord	google
salt	body	concept	grandma	partnership	midwife	pope	posts	debts	profile
sugar	doctors	sexuality	dead	together	eager	Catholics	2014	plant	sensible
tastes	baby	insight	party[pol.]	friendship	wish	minister	reason	psalm	shop
recipes	months	evil	body	married	finally	RCC	Koran	account	price
tomato	birth	Moses	quote	feelings	glad	church	2013	peace	ebay

TABLE IV
EXAMPLE: HIERARCHICAL ORDER OF CLUSTER DESCRIPTIONS. THE CLUSTER IN THE TOP ROW IS SPLIT INTO THE CLUSTERS IN THE LAST 2 ROWS. (WORD FREQUENCIES IN PARENTHESIS, TRANSLATED FROM GERMAN)

	god(335), woman(253), man(251), live(223), jesus(195), love(190), **sex(190)**, question(184), human(183), marriage(156), ...
↪	man(185), **sex(185)**, woman(141), god(109), live(109), women(97), mother(95), topic(94), human(94), problem(89), ...
↪	god(226), live(114), woman(112), love(107), jesus(107), question(96), human(89), god's(89), do(82), marriage(75), ...

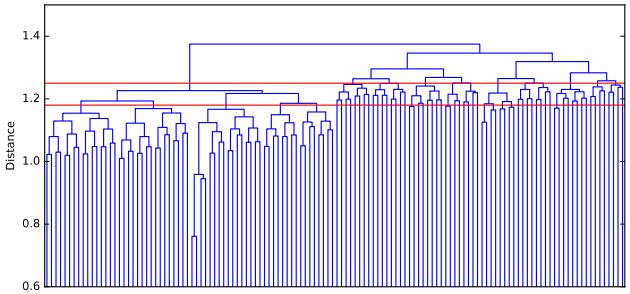

Fig. 2. Hierarchical Clustering of the Online Discussion Dataset. We can derive different flat clusterings by pruning the tree at a specified height, for instance we obtain 10 clusters if we cut at $y = 1.25$ or 100 clusters if we cut at 1.18, as indicated by the horizontal lines.

been closed or moved by a moderator. Topics 9 and 10 are mostly incoherent. Topic 10 seems like a mixture of unrelated topics, the terms 'facebook', 'google', 'profile' and 'ebay' could belong together. Maybe pruning the hierarchy to 100 topics is too harsh and we need more clusters to separate these clusters into coherent subsets. Indeed after splitting the cluster twice, we obtain a cluster with top-keywords 'data','facebook','profile','google','Acta', 'ipaddress', which is more coherent.

This illustrates an interesting property of our approach: We can interactively make topics more specific or more general. We show another instance in Table IV where we see a cluster containing discussions about relationships, love, and marriage. When we split the cluster into its two subclusters, we obtain one cluster that addresses questions of sexuality, while the other does not. Also the second subcluster seems to put more focus on religious aspects of relationships.

In conclusion, we have seen that document embeddings are useful for structuring a text collection and deriving insightful topic descriptions in a deterministic and reproducible fashion. Particularly the possibility to interactively explore the collection and change the granularity of the clusters without expensive computations is appealing.

VI. CONCLUSION AND OUTLOOK

We have presented an algorithm for learning low-rank representations based on regularization, more specifically accelerated proximal gradient descent. The regularizer proposed in this paper, the weighted nuclear norm with the new weighting scheme, $||X||_{\varepsilon}$, allows us to compute representations that approximate the original high-dimensional data with good quality while also being low-dimensional. We demonstrated this in a numerical study where we compared different regularizers.

Then we used our approach to train document embeddings that capture similarities, which we demonstrated using a labeled dataset. We were able to benefit from the distance measure based on embeddings in a digital humanities study, where we extracted coherent and interpretable topics from a collection of online discussions in religious online communities. Our method can be used to interactively explore a document collection and the user can vary the granularity of the topics presented. In the next phase of the digital humanities project, we want to apply similar analysis to the graph of interactions between users, for instance in order to detect communities similar to Yang and Leskovec [8].

Future research should also further investigate the relation between the local minimums of non-convex objectives and the global minimums of convex objectives like Nuclear norm regularization. Empirical studies, including this work, suggest that the local minimums obtain good loss values. It remains to see if there are provable guarantees for the factored models. The weighted nuclear norm allows us to gradually increase the amount of non-convexity in the regularizer. This may allow us to design iterative algorithms with good convergence properties.

Furthermore the investigation of methods for optimization on manifolds seems like a research path worthwhile taking, particularly if we can avoid computing singular value decomposition every iteration.

REFERENCES

[1] Q. Le and T. Mikolov, "Distributed Representations of Sentences and Documents," in *International Conference on Machine Learning - ICML 2014*, vol. 32, 2014, pp. 1188–1196.

[2] S. Deerwester, S. T. Dumais, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, no. 6, pp. 391–407, 1990.

[3] R. Mazumder, T. Hastie, and R. Tibshirani, "Spectral Regularization Algorithms for Learning Large Incomplete Matrices." *Jmlr*, vol. 11, pp. 2287–2322, 2010.

[4] B. Haeffele, E. Young, and R. Vidal, "Structured Low-Rank Matrix Factorization: Optimality, Algorithm, and Applications to Image Processing," in *ICML 2014*, vol. 32, 2014, pp. 2007–2015.

[5] S. Hess, K. Morik, and N. Piatkowski, "The PRIMPING routine—Tiling through proximal alternating linearized minimization," *Data Mining and Knowledge Discovery*, pp. 1–42, 2017.

[6] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning*, corrected ed. Springer, jul 2003.

[7] S. Arora, R. Ge, and A. Moitra, "Learning topic models - Going beyond SVD," in *Proceedings of the Annual IEEE Symposium on Foundations of Computer Science*, 2012, pp. 1–10.

[8] J. Yang and J. Leskovec, "Overlapping community detection at scale: A Nonnegative Matrix Factorization Approach," *Sixth ACM international conference on Web search and data mining*, p. 587, 2013. [Online]. Available: http://dl.acm.org/citation.cfm?id=2433396.2433471

[9] A. Menon and C. Elkan, "Link prediction via matrix factorization," *Ecml Pkdd 2011*, vol. 6912, pp. 437–452, 2011.

[10] T. Mikolov, K. Chen, G. Corrado, J. Dean, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in *Advances in Neural Information Processing Systems*, vol. abs/1310.4, 2013, pp. 1–9.

[11] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: online learning of social representations," *Kdd*, pp. 701–710, 2014.

[12] S. S. Keerthi, T. Schnabel, and R. Khanna, "Towards a Better Understanding of Predict and Count Models," *Arxiv*, pp. 1–17, 2015. [Online]. Available: http://arxiv.org/abs/1511.02024

[13] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, *Stochastic Approximation approach to Stochastic Programming*, 2009, vol. 19.

[14] F. Morin and Y. Bengio, "Hierarchical probabilistic neural network language model," *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pp. 246–252, 2005.

[15] X. Lin and G. Wei, "Accelerated reweighted nuclear norm minimization algorithm for low rank matrix recovery," *Signal Processing*, vol. 114, no. 11171252, pp. 24–33, 2015.

[16] C. Lu, J. Tang, S. Yan, and Z. Lin, "Generalized nonconvex nonsmooth low-rank minimization," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 25, no. 2, pp. 4130–4137, 2014.

[17] G. A. Watson, "Characterization of the subdifferential of some matrix norms," *Linear Algebra and Its Applications*, vol. 170, no. C, pp. 33–45, 1992.

[18] A. Beck and M. Teboulle, "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[19] Y. Nesterov, "A method of solving a convex programming problem with convergence rate O(1/kˆ2)," in *Soviet Mathematics Doklady*, vol. 27, no. 2, 1983, pp. 372–376.

[20] T. Joachims, *Learning to Classify Text using Support Vector Machines*, ser. Kluwer International Series in Engineering and Computer Science. Kluwer, 2002, vol. 668.

[21] R. M. Larsen, "Lanczos bidiagonalization with partial reorthogonalization," *DAIMI Report Series*, vol. 27, no. 537, 1998.

[22] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Minimization," *Optimization Online*, vol. 52, no. 3, pp. 1–33, 2007.

[23] M. Lgorzata Bogdan, E. Van Den Berg, W. Su, and E. J. Can Es, "Statistical Estimation and Testing via the Sorted 1 Norm," pp. 1–46, 2013.

[24] R. Sun and Z. Q. Luo, "Guaranteed Matrix Completion via Non-Convex Factorization," *IEEE Transactions on Information Theory*, vol. 62, no. 11, pp. 6535–6579, 2016.

[25] R. Ge, J. D. Lee, and T. Ma, "Matrix Completion has No Spurious Local Minimum," no. Nips, pp. 1–9, 2016.

[26] C. D. Sa, K. Olukotun, and C. Ré, "Global Convergence of Stochastic Gradient Descent for Some Non-convex Matrix Problems," no. 2, 2015. [Online]. Available: https://arxiv.org/pdf/1411.1134.pdf

[27] B. Vandereycken, "Low-Rank Matrix Completion by Riemannian Optimization," *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1214–1236, 2013. [Online]. Available: https://doi.org/10.1137/110845768

[28] N. Boumal, P.-A. Absil, and C. Cartis, "Global rates of convergence for nonconvex optimization on manifolds," pp. 1–31, 2016.

[29] U. Shalit, D. Weinshall, and G. Chechik, "Online Learning in the Manifold of Low-Rank Matrices," *NIPS*, pp. 1–9, 2010.

[30] T. M. Sanjeev Arora, Yingyu Liang, "A simple but tough-to-beat baseline for sentence embeddings," in *ICLR 2017*, 2017, pp. 1–16.

[31] A. Caliskan, J. J. Bryson, and A. Narayanan, "Semantics derived automatically from language corpora contain human-like biases," *Science*, vol. 356, no. 6334, pp. 183–186, 2017.

[32] S. Robertson, "Understanding inverse document frequency: on theoretical arguments for IDF," *Journal of Documentation*, vol. 60, no. 5, pp. 503–520, 2004.

[33] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *ICLR 2017*, 2016.

[34] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, vol. 3, no. 4-5, pp. 993–1022, 2003.