Document Clustering via Matrix Representation

Xufei Wang
Arizona State University
Tempe, AZ 85287, USA
Email:xufei.wang@asu.edu

Jiliang Tang
Arizona State University
Tempe, AZ 85287, USA
Email:jiliang.tang@asu.edu

Huan Liu
Arizona State University
Tempe, AZ 85287, USA
Email:huan.liu@asu.edu

***Abstract*—The Vector Space Model (VSM) is widely used to represent documents and web pages. It is simple and easy to deal computationally, but it also oversimplifies a document into a vector, susceptible to noise, and cannot explicitly represent underlying topics of a document. A matrix representation of document is proposed in this paper: rows represent distinct terms and columns represent cohesive segments. The matrix model views a document as a set of segments, and each segment is a probability distribution over a limited number of latent topics which can be mapped to clustering structures. The latent topic extraction based on the matrix representation of documents is formulated as a constraint optimization problem in which each matrix (i.e., a document) A_i is factorized into a common base determined by non-negative matrices L and $R^\top$, and a non-negative weight matrix M_i such that the sum of reconstruction error on all documents is minimized. Empirical evaluation demonstrates that it is feasible to use the matrix model for document clustering: (1) compared with vector representation, using matrix representation improves clustering quality consistently, and the proposed approach achieves a relative accuracy improvement up to 66% on the studied datasets; and (2) the proposed method outperforms baseline methods such as k-means and NMF, and complements the state-of-the-art methods like LDA and PLSI. Furthermore, the proposed matrix model allows more refined information retrieval at a segment level instead of at a document level, which enables the return of more relevant documents in information retrieval tasks.**

***Keywords*-Document Clustering, Document Representation, Matrix Representation, Non-Negative Matrix Approximation**

I. Introduction

Document representation is fundamental for data management, information filtering, information retrieval, indexing, classification, and clustering tasks. The Vector Space Model (VSM) represents a document as a vector of terms (or phrases) in which each dimension corresponds to a term (or a phrase). An entry of a vector is non-zero if the corresponding term (or phrase) occurs in the document. A significant progress has been made with the vector space model in many applications. However, it has limitations due to its oversimplification of a document to a term vector. For example, long documents usually contain richer information than short ones, but long documents represented with high-dimensional vectors result in calculations of document similarities that are susceptible to noise. Also one cannot explicitly represent topics in the vector space model.

Documents, especially long ones, often contain multiple topics and are usually organized into one or more logically cohesive segments. A concise statement needs extra illustration or support materials to make it understandable; relevant items are organized into sentences or paragraphs to facilitate comprehension, etc. These observations are supported by numerous examples. For example, a typical news article contains basic elements such as how, who, what, when, where, and why, and sometimes background information and the impact of the event are included. A scientific paper usually contains several sections with different emphasis. Below we show a blog post on TUAW which is related to iPad[1]. The first paragraph of the post introduces an application of the iPad, followed by comments in the second paragraph, and the third paragraph is a factual statement. Apparently, each paragraph covers a specific perspective but all together are related to iPad.

> *Behold the power of iPad. It can charm even the most non-technological of artists. This video highlights a Finnish a cappella group (i.e., a chorus that normally sings without accompaniment, creating all harmonies through voice) who have finally gotten their hands on iPads. Here, they rock out to Madonna's "Material Girl" using iPad-based instrument applications.*
>
> *It's a lovely little video and a nice demonstration of how beautifully the iPad has evolved from a basic tablet into an artistic medium. Chorus member Jani Halme tells TUAW, "Naturally, we shot this using iPhones."*
>
> *The iPad debuted in Finland just a week ago.*

We explore a Matrix Space Model (MSM) to see if it can extend what the Vector Space Model can achieve. Specifically, we propose to *model a document as a matrix* instead of a vector. The two dimensions of a matrix are terms and segments which represent logically independent parts in a document. The underlying assumption is that each (long) document can be divided into more than one *segment* with each corresponding to a few latent topics. Each segment is still represented as a bag of words. Thus, the Matrix Space Model is an extension of the Vector Space Model.

[1]http://www.tuaw.com/2010/12/16/found-footage-ipads-take-the-acapella-out-of-the-girl-group/

However, it differs from the Vector Space Model in that each document is a distribution over latent topics instead of words or phrases. The advantages of the matrix space representation are listed as follows:

- Interpretation is easier by representing a document as a set of cohesive segments, corresponding to a few topics. Some (small) topics will not be oblivious among the terms when a document is viewed as a bag of words. This representation provides a better understanding towards documents by utilizing segmentation techniques.
- Achieving a finer granularity in data management on segments. Long documents with many terms can become obstinate results with vector space representation [1]. Indexing on cohesive segments could mitigate this phenomenon and might be able to return more specific information in terms of segments.
- Being flexible to assign more than one class label to a document as the segments within a document can be assigned with different class labels.

Clustering is an effective way to help deal with large number of documents that are being produced in various real-world applications. Document clustering has been applied to categorization, summarization, and information navigation tasks. However, classic clustering methods are designed to handle data in forms of vectors. When a document is represented as a matrix, new challenges arise for existing document clustering algorithms. We propose a new method for document clustering with matrix representation. Before we assign each document to one or more clusters (disjoint or overlapping clustering), we first extract the latent topics of a corpus. The latent topic extraction is achieved by a non-negative matrix approximation technique that minimizes the sum of reconstruction error on all documents. Segments, instead of documents, with similar distributions over latent topics naturally form clusters. The document could be assigned a most probable label based on the segment labels, or we obtain overlapping clustering by treating each segment label as one of the corresponding document labels.

Document Modeling in a Graphic Interpretation The proposed procedure is illustrated in Figure 1. A document d_i contains multiple segments $(s_1, s_2, \ldots, s_c)$; each segment is a probabilistic distribution over l latent topics $(t_1, t_2, \ldots, t_l)$; and finally latent topics are mapped to k clusters $(c_1, c_2, \ldots, c_k)$ via clustering algorithms. To the best of our knowledge, this model is different from other data representation of documents. Segmentation is a well established realm, thus is not the focus of this work. A major task of this work is to estimate the probabilities between segments and latent topics (topic extraction as shown in Figure 1).

The rest of the paper is organized as follows. We define the problem formally in Section II, followed by the topic extraction in Section III. We build a connection between the proposed approach with Non-negative Matrix Factorization (NMF) techniques in Section IV. Empirical evaluation is presented in Section V with discussion. We summarize the related work in Section VI. The research is concluded in Section VII with some future work.

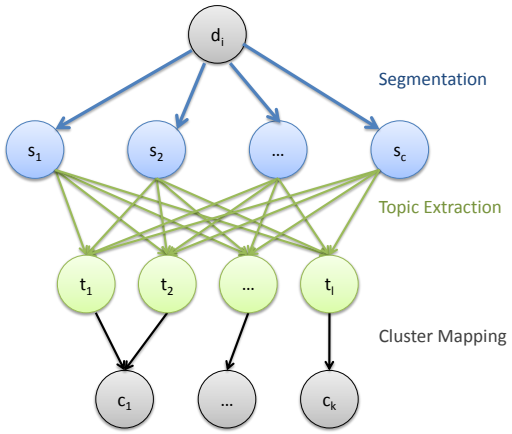

Figure 1. Graph Interpretation

II. Problem Formulation

The studied problem is divided into three components: matrix representation, latent topic extraction, and clustering.

Firstly, we represent documents as matrices leveraging well established segmentation techniques. Let $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$ be a corpus with n documents, Thus, a document d_i is denoted as a matrix $A_i \in \mathcal{R}^{r \times c}$ $(1 \leq i \leq n)$, where r and c correspond to the numbers of terms and segments of the corpus. It is also possible to obtain an arbitrary number of segments for different documents, e.g., longer documents could have more segments than short documents. We encode matrices A_i by TF-IDF weighting such that matrices A_i are non-negative, i.e., $A_i \geq 0$.

Secondly, the latent topic extraction is accomplished by a matrix approximation method. Given two user specified parameters ℓ_1 and ℓ_2 (The smallest reconstruction error is obtained when $\ell_1 = \ell_2$), we want to compute two non-negative basis matrices $L \in \mathcal{R}^{r \times \ell_1} (L \geq 0)$ and $R \in \mathcal{R}^{c \times \ell_2} (R \geq 0)$, and non-negative matrices $M_i \in \mathcal{R}^{\ell_1 \times \ell_2} (M_i \geq 0)$, such that $LM_iR^\top$ is a good approximation for matrix A_i. Computing optimal matrices L, R, and M_i can be formulated as follows,

$$\begin{array}{rcl} \min\limits_{L \in \mathcal{R}^{r \times \ell_1}} & : & L \geq 0 \\ R \in \mathcal{R}^{c \times \ell_2} & : & R \geq 0 \\ M_i \in \mathcal{R}^{\ell_1 \times \ell_2} & : & M_i \geq 0 \end{array} \quad \sum_{i=1}^{n} \|A_i - LM_iR^\top\|_F^2, \quad (1)$$

where $\|\cdot\|_F$ represents the Frobenius norm of a matrix. We can interpret non-negative matrices L and R as jointly defining a lower dimensional space (if ℓ_1 and ℓ_2 are smaller than r and c), and each M_i is the low rank representation of

the original matrices. Specifically, in the case of document clustering, A_i represents the i-th document, matrices L and $R^\top$ define the latent topic space for all data points, and M_i is the mixture weights of a document over all latent topics.

Since L and R are common for all matrices, two documents that are *similar* in the original space are expected to be *similar* in the lower dimensional space. For each document d_i, LM_i represents the posterior probability distribution of each term in a document belonging to latent topics, and $M_iR^\top$ represents the posterior probability distribution of each segment in a document belonging to latent topics.

The above formulation takes a similar form of Singular Value Decomposition (SVD) except that matrices L, M_i and R are non-negative and M_i are usually not diagonal. And it is a generalization of Non-Negative Matrix Factorization (NMF) because L or R are usually not identity matrices. The relation between our formulation and Non-negative Matrix Factorization (NMF) will be further discussed in Section IV.

Thirdly, we obtain document clustering by creating a mapping from the latent topics to clusters via any clustering algorithms (in this paper, we adopt k-means). As an alternative, it is also possible to treat each topic as a cluster if the number of latent topics ℓ is equal to the number of clusters k. In the scope of the paper, we use the first strategy under the assumption that several latent topics could be semantically close thus belong to one cluster. Let $\tilde{d}_{ij}$ be the probability distribution of j-th column of i-th document over ℓ latent topics, there are two ways to assign documents to clusters: 1) obtain *disjoint clustering* by applying k-means to documents. 2) obtaining *overlapping clustering* by applying k-means to segments directly. They can be formulated as follows,

$$\begin{aligned} &\min_k \sum_i \|\tilde{d}_{i\cdot} - centroid(c)\|^2, \quad or \\ &\min_k \sum_{ij} \|\tilde{d_{ij}} - centroid(c)\|^2, \end{aligned} \tag{2}$$

where $\tilde{d}_{i\cdot} = \sum_j \tilde{d}_{ij}$ represents the probability distribution of a document over ℓ latent topics, and $centroid(c)$ represents the centroid of cluster c.

Segmentation and cluster mapping can be obtained by well established methods. Thus, the problem boils down to solving latent topic extraction via Non-negative Matrix Approximation (NMA).

III. Latent Topic Extraction

We propose to solve the problem defined in Eq. (1) by an alternating least square approach. It should be noted that matrix factorization problems are usually not convex, thus there are no guarantees to find optimal solutions. We investigate how to obtain a local optimal solution, and by running multiple times with different initializations, aiming to achieve one close to the optimal solution in practice.

We solve the problem in two steps. We first define non-negative auxillary matrices $B_i (1 \le i \le n)$, such that each A_i can be factorized as follows,

$$A_i = LB_i, \ i = 1, 2, \ldots, n. \tag{3}$$

Then, in the second step, each matrix B_i is further factorized into two matrices M_i and R which can be shown by,

$$B_i = M_iR^\top, \ i = 1, 2, \ldots, n. \tag{4}$$

Both sub-problems defined in Eq. (3) and Eq. (4) are non-convex but can be solved by an alternating scheme: for Eq. (3), we first fix L and compute B_i, then update L by fixing all matrices B_i. The procedure is repeated until convergence. The same procedure can be applied to Eq. (4).

Fix L, compute B_i. When L is fixed, B_i can be computed by solving a series of non-negative least square problems. Specifically, let a_i^l and b_i^l be the l-th columns of matrices A_i and B_i, respectively. Then b_i^l can be computed by solving the following non-negative least square problem,

$$\min_{b_i^l} \ \|Lb_i^l - a_i^l\|_2^2, \ s.t., \ b_i^l \ge 0, \tag{5}$$

where $l = 1, 2, \ldots, c$, $i = 1, 2, \ldots, n$, and $b_i^l \ge 0$ denotes that all entries in b_i^l are non-negative.

Fix B_i, compute L. After all matrices B_i are obtained, we fix all B_i and compute matrix L by solving Eq. (6),

$$\min_L \ \sum_{i=1}^n \|A_i - LB_i\|_F^2 \tag{6}$$

It follows that,

$$\begin{aligned} &\sum_{i=1}^n \|A_i - LB_i\|_F^2 \\ &= Tr\left(\sum_{i=1}^n (A_iA_i^\top + LB_iB_i^\top L^\top - 2LB_iA_i^\top)\right) \end{aligned} \tag{7}$$

where $Tr(\cdot)$ represents the trace of a matrix. By removing constant term $Tr(A_iA_i^\top)$ in Eq. (7), denoting $M = \sum_{i=1}^n B_iB_i^\top$, $N = \sum_{i=1}^n B_iA_i^\top$, and adding a constant term $N^\top M^{-1}N$. The objective function in Eq. (7) can be rewritten as follows,

$$\begin{aligned} &Tr\left(LML^\top - 2LN + N^\top M^{-1}N\right) \\ &= Tr\left(LM^{\frac{1}{2}} - N^\top M^{-\frac{1}{2}}\right)\left(LM^{\frac{1}{2}} - N^\top M^{-\frac{1}{2}}\right)^\top \\ &= \|LM^{\frac{1}{2}} - N^\top M^{-\frac{1}{2}}\|_F^2 \\ &= \|M^{\frac{1}{2}}L^\top - M^{-\frac{1}{2}}N\|_F^2 \end{aligned} \tag{8}$$

It should be noted that matrix M is positive semidefinite. Given any column vector x,

$$\begin{aligned} x^\top Mx &= x^\top \sum_{i=1}^n B_iB_i^\top x \\ &= \sum_{i=1}^n (B_i^\top x)^\top B_i^\top x \ge 0 \end{aligned} \tag{9}$$

Algorithm 1 Latent Topic Extraction via Approximation of Matrices

Input: data A_i, ($1 \leq$ i $\leq$ n), dimensions of matrix M_i: ℓ_1 and ℓ_2, maximum iterations: max_iter.
Initialize $L = L_0$.
repeat
  for $i = 1$ **to** n **do**
    compute B_i by solving Eq. (5).
  end for
  compute L by solving Eq. (12).
until convergence
Initialize $R = R_0$.
repeat
  compute M_i and R by solving Eq. (13).
until convergence

Thus, $M^{\frac{1}{2}}$ and $M^{-\frac{1}{2}}$ can be obtained by eigen decomposition of M

$$\begin{aligned} M^{\frac{1}{2}} &= VD^{\frac{1}{2}}V^\top \\ M^{-\frac{1}{2}} &= VD^{-\frac{1}{2}}V^\top \end{aligned} \tag{10}$$

where V is the orthogonal matrix consisting of eigenvectors of M, and D is a diagonal matrix whose entries are eigenvalues. Plugging $M^{\frac{1}{2}}$ and $M^{-\frac{1}{2}}$ into Eq. (8), we obtain,

$$\|D^{\frac{1}{2}}V^\top L^\top - D^{-\frac{1}{2}}V^\top N\|_F^2 \tag{11}$$

Given matrix $L = (\ell_1^\top, \ell_2^\top, \ldots, l_r^\top)^\top$, each row can be computed by solving the following least square problem,

$$\min_{l_i} \ \|D^{\frac{1}{2}}V^\top l_i^\top - (D^{-\frac{1}{2}}V^\top N)_i^\top\|_2^2, \ s.t., \ l_i \geq 0 \tag{12}$$

where $i = 1, 2, \ldots, r$, l_i represents the i-th row of matrix L. Matrices L and B_i are updated iteratively until they converge.

Eq. (4) can be solved in a similar procedure by factorizing each B_i into M_i and R. The details are omitted and only the key steps are presented in Eq. (13),

$$\begin{aligned} &\min_{M_i} \ \|RM_i^\top - B^\top\|_F^2, \ s.t., \ M_i^\top \geq 0 \\ &\min_{R} \ \|P^{\frac{1}{2}}R^\top - P^{-\frac{1}{2}}Q\|_F^2 \ s.t., \ R \geq 0 \end{aligned} \tag{13}$$

where $P = \sum_{i=1}^n M_i^\top M_i$, $Q = \sum_{i=1}^n M_i^\top B_i$.

Finally, we obtain the following two-sided approximations for the original matrices A_i which can be shown as,

$$A_i \approx LM_iR^\top, \ i = 1, 2, \ldots, n, \tag{14}$$

where matrices L, M_i, and R are all non-negative. Above procedures are summarized in Algorithm (1).

Given a dataset with n instances, we need to solve nc and r non-negative least square problems for computing Eq. (5) and Eq. (12), respectively. Similarly, we need $n\ell_1$ and c non-negative least square problems to solve for computing Eq. (13), respectively. Thus, the computational complexity of Algorithm (1) is $O\left(k_1(nc + r) + k_2(n\ell_1 + c)\right)$, where k_1 and k_2 are the number of iterations for solving Eq. (3) and Eq. (4), respectively.

IV. Connection to Non-negative Matrix Factorization (NMF)

From the perspective of matrix factorization, we obtain a two sided non-negative matrix factorization via NMA. Thus, it is a special form of Non-negative Matrix Factorization. For simplicity, let $Z_i = A_i - LM_iR^\top$ $(1 \leq i \leq n)$. We can rewrite Eq. (1) as follows,

$$\begin{aligned} &\sum_{i=1}^n \|A_i - LM_iR^\top\|_F^2 \\ &= \sum_{i=1}^n \|Z_i\|_F^2 \\ &= \sum_{i=1}^n Tr(Z_i {Z_i}^\top) \\ &= Tr\begin{pmatrix} Z_1{Z_1}^\top & 0 & \ldots & 0 \\ 0 & Z_2{Z_2}^\top & \ldots & 0 \\ \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & \ldots & Z_n{Z_n}^\top \end{pmatrix} \\ &= Tr\begin{pmatrix} Z_1 & 0 & \ldots & 0 \\ 0 & Z_2 & \ldots & 0 \\ \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & \ldots & Z_n \end{pmatrix}\begin{pmatrix} {Z_1}^\top & 0 & \ldots & 0 \\ 0 & {Z_2}^\top & \ldots & 0 \\ \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & \ldots & {Z_n}^\top \end{pmatrix} \\ &= Tr(\mathcal{A} - \mathcal{L}\mathcal{M}\mathcal{R}^\top)(\mathcal{A} - \mathcal{L}\mathcal{M}\mathcal{R}^\top)^\top \\ &= \|\mathcal{A} - \mathcal{L}\mathcal{M}\mathcal{R}^\top\|_F^2, \end{aligned} \tag{15}$$

The matrices $\mathcal{A}$, $\mathcal{L}$, $\mathcal{M}$, and $\mathcal{R}$ are given as follows,

$$\begin{aligned} \mathcal{A} &= \text{Diag}\left(A_1, A_2, \ldots, A_n\right) \\ \mathcal{M} &= \text{Diag}\left(M_1, M_2, \ldots, M_n\right) \\ \mathcal{L} &= \text{Diag}(L, L, \ldots, L) \\ \mathcal{R} &= \text{Diag}(R, R, \ldots, R) \end{aligned} \tag{16}$$

where $Diag(\ldots)$ represents a block diagonal matrices whose main diagonal blocks are matrices and other entries are all zeros. Apparently, matrices $\mathcal{L}$, $\mathcal{M}$ and $\mathcal{R}^\top$ are all non-negative. Although Eq. (15) takes the same form of Tri-Factorization [2], the difference between NMA and NMF are summarized below,

- Representations are different. Each document is a represented as a matrix in Eq. (1), however, for Non-negative Matrix Factorization each document is represented as a vector.
- $\mathcal{L}$ and $\mathcal{R}^\top$ are block diagonal matrices but not for Non-negative Matrix Factorization.

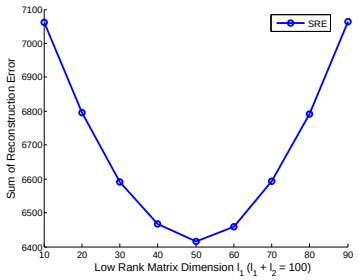

Figure 2. Reconstruction Error w.r.t ℓ_1/ℓ_2

V. Empirical Evaluation

We will introduce the datasets used in experiments, examine how clustering accuracy is affected by varying the number of latent topics and the number of segments in a document, demonstrate the advantages of adopting a matrix representation, and compare the proposed NMA approach with representative document clustering methods to understand their pros and cons.

A. The Effect of ℓ_1/ℓ_2

Theoretical analysis on matrix approximation shows that the sum of reconstruction error is minimized [3] when $\ell_1 \approx \ell_2$. In the case of non-negative matrix approximation, we verify the theoretic result using a synthetic dataset. It consists of 500 matrices with dimensions 100 by 100 and the entries of the matrices being between 0 and 255. The purpose is to show the reconstruction error with respect to dimensions ℓ_1 and ℓ_2 of matrices M_i. The sum of reconstruction error is defined as follows,

$$SRE = \sum_{i=1}^{n} \|A_i - LM_iR^\top\|_F^2 \tag{17}$$

this phenomenon is verified on the dataset. We fix the summation of ℓ_1 and ℓ_2 to 100. As shown in Figure 2, the minimum reconstruction error is achieved when $\ell_1 \approx \ell_2$. Thus, in the following experiments, we set $\ell_1 = \ell_2 = \ell$, where ℓ is called *the number of latent topics*.

B. Text Corpus Introduction

Three widely used real text corpora are used to evaluate the performance of NMA with four representative document clustering methods.

20Newsgroup[2] is a collection of documents across 20 different newsgroups. It contains 19,997 documents that are roughly, evenly distributed in 20 groups. Short documents with less than 1,000 characters are removed from the dataset, resulting in 6,038 documents distributed in 20 groups. The maximum, mean, and minimum groups have 506, 301, and 27 documents, respectively. Except for one small cluster with 27 instances, other documents are roughly evenly distributed to 19 clusters. Another important observation of the dataset is that some of the clusters share a large number of keywords.

Table I
Text Corpus Statistics

Dataset	Newsgroup	Reuters	Classic
# of Docs	19,997	21,578	3,893
# of Docs Used	6,038	1,964	1,486
# of Clusters	20	135	3
# of Clu. Used	20	26	3
# of Words	60,161	13,082	10,935
Max Cluster Size	506	471	626
Avg Cluster Size	301	71	495
Min Cluster Size	27	12	385

Reuters-21578[3] is a testbed for clustering purposes since the labels are manually specified. We remove the short documents with less than 1,000 characters. Documents with multiple labels are also excluded. Finally 1,964 documents are obtained and distributed in 26 clusters. The maximum, mean, and minimum cluster have 471, 71, and 12 documents, respectively. This dataset is highly unbalanced in terms of document distribution in clusters.

Classic[4] is a text corpus on scientific papers. It has 1,400 *Cranfield* documents from aeronautical system papers, 1,033 *medline* documents from medical journals, and 1,460 *cisi* documents from information retrieval papers. Documents less than 1,000 characters are removed and finally we get a dataset with 1,486 documents.

The datasets are preprocessed to remove stop words, stem terms by the Porter Stemming algorithm, and terms are then weighted by TF-IDF weighting scheme. Different text segmentation techniques can be applied to extract segments from documents. The linear text segmentation by Choi et al. [4] is used in our experiments because the studied documents are not very long, thus it is more appropriate to segment based on sentences, and technically, any segmentation approach can be applied to extract segments. We do not attempt to propose a new segmentation algorithm, nor do we attempt to compare which segmentation approach is superior in the rest of the paper.

Short documents with less than 1,000 characters including space, stop words and etc are excluded based on two observations: short documents contain too few terms after removing stop words, spaces, header and footer, e.g., some documents may have few words or blank at all, thus these extremely short documents might become less meaningful for clustering tasks; short documents containing too few sentences are difficult to be divided into meaningful segments. The segmentation algorithms are typically running clustering algorithms at different resolutions such as sentence, paragraph. Thus, the documents should not be too short

[2]http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html

[3]http://archive.ics.uci.edu/ml/datasets/Reuters-21578+Text+Categorization+Collection

[4]ftp://ftp.cs.cornell.edu/pub/smart/

when applying segmentation. Finally, the average numbers of words in the three datasets 20Newsgroup, Reuters-21578, and Classic are 113, 97, and 108, respectively.

In the following experiments, we use $k(2 \leq k \leq 5)$ and ℓ to denote the number of clusters and the number of latent topics, respectively. Given a cluster number k, the number of latent topics is set to at least k (i.e., $\ell \geq k$). Experiments with different k and ℓ values are extensively carried out. Given the number of clusters k, the number of latent topics ℓ, and a dataset $\mathcal{D}$, we evaluate the proposed clustering algorithms in the following three steps,

- **Dataset Generation:** we randomly select k classes with the documents belonging to the k classes from dataset $\mathcal{D}$ to form a new dataset $\mathcal{D}_k$ which is a subset of $\mathcal{D}$,
- **Evaluation:** we run the proposed clustering algorithm or other baseline algorithms 10 times with different initializations on $\mathcal{D}_k$ by setting the number of clusters to k, then obtain the mean accuracy, and
- **Repeat:** we repeat above steps 50 times and report the mean accuracy for pre-specified parameters k and ℓ.

We apply the exact same procedures on the baseline methods which will be introduced next since clustering algorithms such as k-means and NMF typically converge to local minima. We do not perform clustering on the whole dataset but on the datasets we constructed that are subsets of the whole dataset.

C. Baseline Methods and Metric

We compare our methods to four methods for document clustering. K-means is to partition n documents into disjoint k clusters such that the inter cluster similarity is minimized, meanwhile the inner cluster similarity is maximized. K-means usually converges to local minima. Non-negative Matrix Factorization (NMF) has recently been successfully applied to document clustering [5], [6]. It computes the probability of each document belonging to each cluster and a document is assigned to the cluster with maximum probability. Similar to k-means, the cluster assignment is usually not optimal. Topic models such as Probabilistic Latent Semantic Indexing (PLSI) and Latent Dirichlet Allocation (LDA) have been used to infer clusters.

We evaluate the clustering quality by accuracy on a real text corpus. Given a clustering, the label of each cluster is determined by the class that dominates this cluster. Denoting $l(c)$ as the label of a cluster c, $l(d_j)$ as the predicted label of the j-th document, the accuracy is defined as follows,

$$ACC = \frac{1}{n} \sum_{c}^{k} \sum_{d_j \in c} \delta(l(d_j), l(c)) \tag{18}$$

D. Latent Topics v.s. Clustering Quality

In practice, the number of latent topics should be bounded by a limited number in a corpus. Fixing the number of clusters k from 2 to 5, we set the number of latent topics

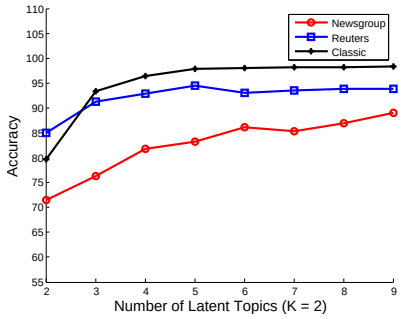

Figure 3. Clustering Quality w.r.t. the Number of Latent Topics (k=2)

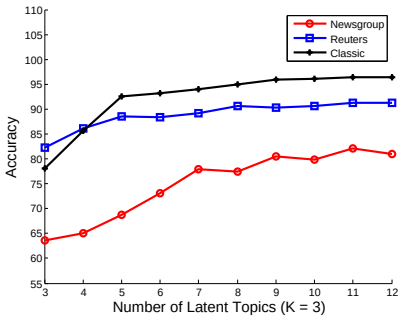

Figure 4. Clustering Quality w.r.t. the Number of Latent Topics (k=3)

from k to $(3k + 3)$ with an increment of 1. For example, when k is set to 2, the latent topics are set to between 2 and 9. Then the average accuracy on 50 runs for each dataset is reported in Figures 3—6. As shown in these figures, the accuracy gets improved as the number of latent topics increases. For example, the average relative improvements are 17.7%, 9.75%, 22.0% when the number of latent topics are set to at least $k + 1$ on the Newsgroup, Reuters, and Classic datasets, respectively.

We also observe another pattern: the accuracy is improved acutely when the latent topics are increased at the first few steps (i.e. $l = k + 1$ or $l = k + 2$), then the improvement becomes flat. When the number of latent topics is set to number of clusters, the performance is not as good as that when a larger number of latent topics is used.

E. Matrix Representation v.s. Vector Representation

The number of segments controls the granularity in modeling a document. Since the studied datasets are not very long, the number of segments are set to between 1 to 6, recalling that a document is represented as a vector when the number of segments is set to 1. To fare to compare, we fix the number of the latent topics and number of clusters but vary the number of segments. The results are reported in Figure 7—10, in which the x-axis represents the number of segments, and y-axis represents the mean accuracy. The four figures correspond to datasets constructed with different numbers of classes, i.e., we construct datasets with 2, 3, 4,

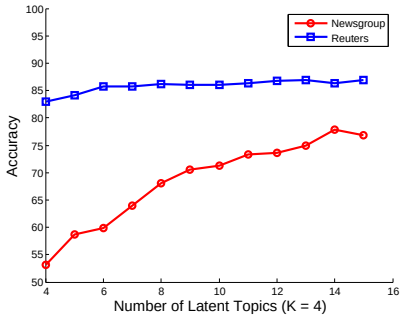

Figure 5. Clustering Quality w.r.t. the Number of Latent Topics (k=4)

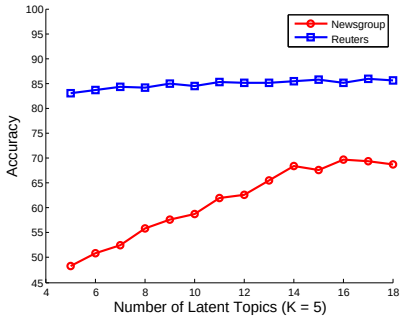

Figure 6. Clustering Quality w.r.t. the Number of Latent Topics (k=5)

and 5 clusters, respectively. The accuracy is averaged on 50 randomly constructed datasets with different initializations, thus it is an decent indicator that reflects how the number of segments affect the clustering quality.

In each figure, the first column represents the utilizing vector representation, while the other columns represent matrix representation. They consistently show that utilizing multiple segments outperforms single segment. We achieve a relative improvement up to 66% on Newsgroup dataset when k and ℓ are set to 4 and 3 (Figure 9), respectively. We also observe another interesting pattern: the improvement peaks when the number of segments is set to 2 or 3 for the studied datasets, then drops a little when a document is spit into more segments. This is partly because the studied documents are rather short thus splitting a document into more segments makes the segments too small to be meaningful. However, this is helpful in practice for selecting the number of segments.

In Table II, we summarize the relative improvement in

Table II
THE RELATIVE IMPROVEMENT WHEN COMPARING MATRIX REPRESENTATION TO VECTOR REPRESENTATION

Dataset	Number of Clusters			
	2	3	4	5
Newsgroup	22.85	52.71	59.07	51.3
Reuters	7.61	13.38	7.55	7.44
Classic	11.74	11.70	-	-

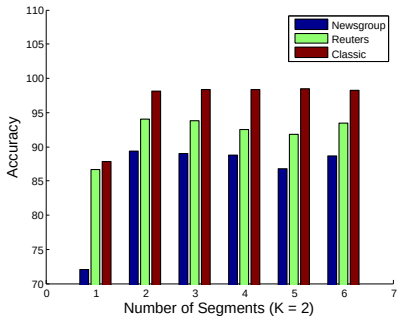

Figure 7. Clustering Quality w.r.t the Number of Segments (k=2)

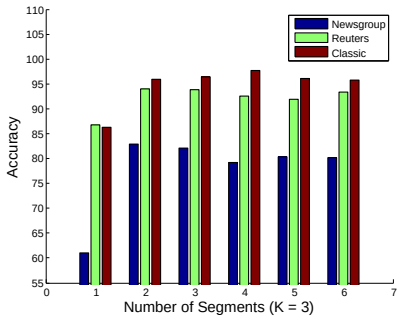

Figure 8. Clustering Quality w.r.t the Number of Segments (k=3)

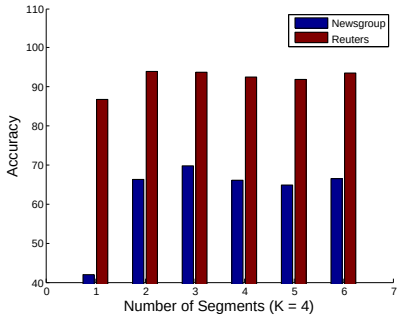

Figure 9. Clustering Quality w.r.t the Number of Segments (k=4)

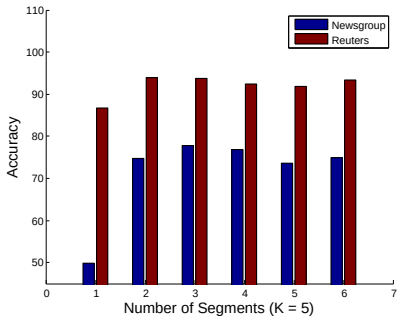

Figure 10. Clustering Quality w.r.t the Number of Segments (k=5)

Table III
ACCURACY PERFORMANCE ON NEWSGROUP

k	k-means	NMF	PLSI	LDA	NMA
2	69.77	72.04	84.87	88.66	**88.97**
3	49.58	60.98	76.39	**84.84**	82.14
4	40.43	49.93	71.64	**81.14**	77.79
5	34.52	41.92	69.91	**78.73**	69.72

terms of accuracy when matrix representation is compared to vector representation. In the table, each entry represents the relative improvement that are averaged on the number of segments ranging from 2 to 6 in the experiments. This table essentially shows the superiority of the matrix presentation over the vector representation of documents.Since Classic dataset only contains 3 classes, two of the corresponding entries are vacant. Although the improvement varies with respect to the number of classes selected, we achieve consistent improvement by applying matrix representation.

Currently, we assign a unique number of segments to all documents in a dataset, but documents in different lengths deserve to be treated differently. For instance, in general, longer documents could be split into more segments and shorter documents might have fewer segments. This study could be interesting and will be exploited as future work.

F. Comparative Study

Tables III, IV, and V show the average accuracy on the three datasets: Newsgroup, Reuters, and Classic, respectively. Each column represents a clustering method. Each entry of the three tables represents the accuracy of the proposed approach and baseline methods that is averaged on 50 randomly generated datasets with the number of classes k pre-specified.

On the Newsgroup dataset (Table III), the proposed method is comparable to the two state-of-the-art methods LDA and PLSI. Compared to k-means and NMF, NMA gains 71.9% and 45.1% relative improvements, respectively. We noticed that the classes in Newsgroup have relatively large numbers of documents, probabilistic models such as LDA and PLSI which utilize the co-occurrence of terms are capable of learning more accurate models than k-means and NMF. It seems that other methods are more susceptible to be affected than LDA by the number of clusters in the datasets.

The performance on the Reuters dataset is summarized in Table IV: k-means performs the worst in all cases, NMF, PLSI, and LDA are close, while NMA method outperforms other methods, especially when the k is set to 2 and 3. On average, NMA gains 20.42%, 7.46%, 18.52%, and 10.37% improvement in accuracy compared with k-means, NMF, PLSI, and LDA, respectively. We notice that for this dataset, the average cluster size is relatively small and the size of a cluster varies significantly as seen in Table I.

As shown in Table V on the Classic dataset, NMA and LDA are comparable, and PLSI follows very close.

Table IV
ACCURACY PERFORMANCE ON REUTERS

k	k-means	NMF	PLSI	LDA	NMA
2	77.68	86.68	79.08	83.33	**94.47**
3	70.37	81.75	72.49	80.95	**91.32**
4	75.96	82.25	77.27	81.57	**86.95**
5	73.85	83.09	73.78	79.11	**85.94**

Table V
ACCURACY PERFORMANCE ON CLASSIC

k	k-means	NMF	PLSI	LDA	NMA
2	76.71	87.88	97.49	97.78	**98.36**
3	77.26	86.29	96.60	**97.57**	96.45

Compared to the other two datasets, the clusters are larger in size and more balanced, and the clusters are very different in content as the documents are from different fields. LDA and PLSI are capable of learning the cluster structure accurately when the corpus is large and cluster boundaries are clear. The performance of NMA is relatively stable.

G. Further Discussions

Compared to the Vector Space Model, the advantages of Matrix Space Representation of documents can be summarized as follows: gaining a finer granularity (segments instead of documents) in information retrieval tasks, helping interpret documents topics, and providing quality document clusterings. However, it also comes with some limitations: storing a matrix requires more space, computing latent topics is not as efficient by matrix approximation as using vectors. But it should also be noticed that since the matrix representation is more sparse than the vector representation (each segment contains fewer terms), the sparsity property can be utilized to improve computational efficiency.

The consistent improvement in accuracy performance by utilizing the matrix representation versus the vector representation demonstrates the superiority of the new data representation though there is a cost to do that. We demonstrate that it is not only feasible but also more preferable to encode documents as matrices instead of vectors. Though this paper use clustering to illustrate the power of the matrix representation, it is also intriguing to examine whether it is worth adapting the matrix representation in other applications such as classification and information retrieval.

VI. RELATED WORK

Document clustering is useful in data management, information retrieval, filtering, and navigation. It is related to several key components: document representation, segmentation, topic models, and clustering algorithms. Next we summarize each component separately.

A. Document Representation

The Vector Space Model (VSM) is widely used in text and web mining, information filtering, information retrieval,

etc. It views a document as a bag of words or phrases which assumes the independence between different terms and ignores the order of the terms in a document. Some extensions of the Vector Space Model are focused on feature construction. For example, combining terms into syntactic phrases [7], using anchor texts to enrich document representation [8], extracting name entities [9], and constructing bag of phrases [10] as new features. However, the VSM representation of documents has some limitations: 1) document similarity is susceptible to noise because of high dimensionality; 2) topics within a document are hidden behind bag of words thus is not intuitive to interpret. 3) obstinate results may hurt user satisfaction in information retrieval applications [1].

Liu et al. propose to represent documents as tensors [11]. All terms including special characters within a document are split into consecutive equal sized substrings, then each character in a substring represents a dimension of the tensors. For example, if the substring length is 3, then a document is represented by a tensor $T^{27\times27\times27}$. Apparently, this model is not storage efficient and substrings are usually invalid words or in even worse cases the semantic meaning of a word could be totally changed if a substring refers to another word.

B. Document Segmentation

Text segmentation is to divide text into multiple meaningful units which is termed as "segments" in this paper. It is a well established area and there are efficient and sophisticated methods for segmentation which is not the focus of this work. Each segment is a relatively cohesive component within a document. Segmentation approaches can be based on words, sentences, or paragraphs. The simplest way of segmentation is to divide a document into several roughly equal-sized components thus is based on words. Choi et al. [4] propose to extract text segmentation based on sentence similarity and the boundary of segments are determined by divisive clustering algorithms. Tagarelli et al. [12] divides the documents into segments based on clustering on paragraphs. This method works for long documents.

C. Topic Models

Topic models assume a document consists of one or more topics which can be treated as clusters. Hofmann et al. [13] proposed Probabilistic Latent Semantic Indexing (PLSI), which models each word in a document as a sample from a mixture model, where the mixture components are multinomial random variables that can be viewed as representations of "topics". Thus each word is generated from a single topic and different words in a document may be generated from different topics. Latent Dirichlet Allocation (LDA) [14], which is also extended from Latent Semantic Indexing (LSI), is currently considered as the state-of-the-art method in text clustering. LDA assumes that each document is a mixture of a small number of topics and that each word's creation is attributed to one of the document's topics. Although both PLSI and LDA view words and documents as mixture of topics, they ignore word order.

D. Document Clustering

Text clustering has a broad application in topic extraction, information retrieval, information filtering, etc. Most popular document clustering approaches are based on the Vector Space Model. Clustering techniques can be roughly divided into two categories: discriminative and generative. The discriminative algorithms optimize an object function to produce an pseudo optimal clustering; whereas, the generative algorithms assumes that the data can be modeled by an underlying distribution such that the parameters of a model could be learned iteratively [15]. Instead of clustering documents, co-clustering which is related to spectral clustering [16] attempts to clustering the documents and words simultaneously [17].

E. Non-negative Matrix Factorization

Non-negative Matrix Factorization (NMF) is a technique for co-clustering the rows and columns of a matrix. It has been applied in document clustering tasks. The basic idea is to find non-negative matrices W and H, such that their product is an approximation of the original matrix X, i.e. $X = WH$. If X is a term document matrix, matrices W and H can be interpreted as a cluster structure and memberships of a document belong to each cluster, respectively. The rank of matrices W and H are often smaller than that of X. Thus, W and H can be thought of as a compressed form for matrix X. However, the factorization of matrices are usually not unique, and the solvers can be roughly classified into three categories: Multiplicative Update Methods [18], Gradient Descendent Methods [19], [5], [20], [21], and Alternating Least Squares Methods [22], [23]. The different NMF techniques and their relationships with other clustering techniques are summarized by Li and Ding [24].

VII. Conclusion and Future Work

In this paper, we propose to represent a document as a matrix in which the two dimensions are terms and segments. A document is then modeled as follows: a document is a set of segments and a segment is a mixture over a limited number of latent topics. Documents are assigned to a cluster if they have similar probability distributions on latent topics. Experimental evaluations show that 1) it is feasible to use the matrix space model for document clustering and we achieve consistent improvement in terms of accuracy when compared with vector representation, 2) the proposed document clustering approach outperforms baseline methods such as k-means and NMF, and 3) the matrix representation complements the vector representation in handling different types of data.

This work also suggests some promising research opportunities. One direction is to examine how precision and recall vary by indexing segments instead of documents in information retrieval. The practical significance of this extension is that we enable the return of relevant segments in a document. Another direction is to adapt existing document clustering methods to accept matrix space representation of documents as input to explore expanded capabilities of document clustering.

VIII. Acknowledgments

This work is, in part, supported by ONR (N000141010091) and NSF (#0812551), and the views are solely of the authors'.

References

[1] M. Radovanović, A. Nanopoulos, and M. Ivanović, "On the existence of obstinate results in vector space models," in *Proceedings of the 33rd annual international ACM SIGIR conference on Research and development in informaion retrieval (SIGIR'10)*, 2010.

[2] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix trifactorizations for clustering," in *The 12th Annual SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'06)*, 2006.

[3] J. Liu, S. Chen, Z. Zhou, and X. Tan, "Generalized low rank approximations of matrices revisited," *IEEE Transactions on Neural Networks*, vol. 21, no. 4, pp. 621–632, 2010.

[4] F. Y. Y. Choi, "Advances in domain independent linear text segmentation," in *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, San Francisco, CA, USA, 2000, pp. 26 – 33.

[5] F. Shahnaz, M. W. Berry, V. P. Pauca, and R. J. Plemmons, "Document clustering using nonnegative matrix factorization," *Information Processing & Management*, vol. 42, no. 2, pp. 373 – 386, 2006.

[6] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval (SIGIR'03)*, 2003.

[7] D. D. Lewis and W. B. Croft, "Term clustering of syntactic phrases," in *Proceedings of the 13th annual international ACM SIGIR conference on Research and development in informaion retrieval (SIGIR'89)*, 1989.

[8] D. Metzler, J. Novak, H. Cui, and S. Reddy, "Building enriched document representations using aggregated anchor text," in *Proceedings of the 32nd annual international ACM SIGIR conference on Research and development in informaion retrieval (SIGIR'09)*, 2009.

[9] G. Kumaran and J. Allan, "Text classification and named entities for new event detection," in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in informaion retrieval (SIGIR'04)*, 2004.

[10] H. Chim and X. Deng, "Efficient phrase-based document similarity for clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 9, pp. 1217–1229, September 2008.

[11] N. Liu, B. Zhang, J. Yan, Z. Chen, W. Liu, F. Bai, and L. Chien, "Text representation: from vector to tensor," in *Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM'05)*, 2005.

[12] A. Tagarelli and G. Karypis, "A segment-based approach to clustering multi-topic documents," in *Text Mining Workshop, SIAM Datamining Conference*, 2008.

[13] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Machine Learning*, vol. 42, pp. 177–196, 2001.

[14] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *The Journal of Machine Learning Research*, vol. 3, pp. 993 – 1022, 2003.

[15] N. O. Andrews and E. A. Fox, "Recent developments in document clustering," Computer Science, Virginia Tech, Tech. Rep. TR-07-35, 2007.

[16] U. von Luxburg, "A tutorial on spectral clustering," Max Planck Institute for Biological Cybernetics, Tech. Rep., 2006.

[17] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *Proceedings of the seventh ACM SIGKDD international conference on knowledge discovery and data mining (KDD'01)*, 2001, pp. 269 — 274.

[18] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[19] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.

[20] C.-J. Lin, "Projected gradient methods for non-negative matrix factorization," *Neural Computation*, vol. 19, no. 10, pp. 2756 – 2779, 2007.

[21] V. P. Pauca, J. Piper, and R. J. Plemmons, "Nonnegative matrix factorization for spectral data analysis," *Linear Algebra and its Applications*, vol. 416, no. 1, pp. 29 – 47, July 2006.

[22] P. Paatero and U. Tapper, "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111 – 126, 1994.

[23] R. Albright, J. Cox, D. Duling, A. N. Langville, and C. D. Meyer, "Algorithms, initializations, and convergence for the nonnegative matrix factorization," North Carolina State University, Tech. Rep. 81706, 2006.

[24] T. Li and C. Ding, "The relationships among various nonnegative matrix factorization methods for clustering," in *IEEE International Conference on Data Mining (ICDM'06)*, 2006.