



ELSEVIER

Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Information Processing and Management xxx (2004) xxx–xxx

**INFORMATION
PROCESSING
&
MANAGEMENT**
www.elsevier.com/locate/infoproman

2 Document clustering using nonnegative matrix factorization ☆

3 Farial Shahnaz ^a, Michael W. Berry ^{a,*}, V. Paul Pauca ^b, Robert J. Plemmons ^b

4 ^a Department of Computer Science, University of Tennessee, Knoxville, TN 37996-3450, USA

5 ^b Department of Computer Science, Wake Forest University, Winston-Salem, NC 27109, USA

Received 24 August 2004; accepted 19 November 2004

8 Abstract

9 A methodology for automatically identifying and clustering semantic features or topics in a heterogeneous text col-
10 lection is presented. Textual data is encoded using a low rank nonnegative matrix factorization algorithm to retain nat-
11 ural data nonnegativity, thereby eliminating the need to use subtractive basis vector and encoding calculations present
12 in other techniques such as principal component analysis for semantic feature abstraction. Existing techniques for non-
13 negative matrix factorization are reviewed and a new hybrid technique for nonnegative matrix factorization is pro-
14 posed. Performance evaluations of the proposed method are conducted on a few benchmark text collections used in
15 standard topic detection studies.

16 © 2004 Elsevier Ltd. All rights reserved.

17 *Keywords:* Nonnegative matrix factorization; Text mining; Conjugate gradient; Constrained least squares

19 1. Introduction

20 Text mining refers to the detection of trends, patterns, or similarities in natural language text. Given a
21 collection of text documents, often the need arises to classify the documents into groups or clusters based
22 on similarity of content. For a relatively small collection, it may be possible to manually perform the par-
23 titioning of documents into specific categories. But to partition large volumes of text, the process would be
24 extremely time consuming. Moreover, automation also greatly reduces the time needed to perform the
25 classification.

* Research supported in part by the Air Force Office of Scientific Research under grant FA49620-03-1-0215, and by the Army Research Office under grant DAAD19-00-1-0540.

* Corresponding author.

E-mail addresses: shahnaz@cs.utk.edu (F. Shahnaz), berry@cs.utk.edu (M.W. Berry), paucavp@wfu.edu (V. Paul Pauca), plemmons@wfu.edu (R.J. Plemmons).

26 When the categories or topics for classification are predefined, the process of classification is considered
27 *supervised*; there are several methods in use that satisfactorily automate the task of supervised classification
28 (Dunham, 2003). However, in absence of any information regarding the nature of the data, the problem of
29 classification becomes much more difficult. For *unsupervised* classification of text data, only one valid
30 assumption can be made, which is that the text collection is completely unstructured. The task then be-
31 comes organizing the documents into a structure based solely on patterns learned from the collection itself.
32 This structure can be *partitional* or *hierarchical* (Dunham, 2003). The hierarchical organization of docu-
33 ments has a tree-like structure with the entire collection situated at the root level. In subsequent levels of
34 the tree, the collection is partitioned into smaller groups and eventually each document is represented as
35 a separate group at the bottom level.

36 If the text collection is given a partitional structure, then the documents in the collection are flatly par-
37 titioned or clustered into groups that are nonoverlapping. The proposed nonnegative matrix factorization
38 (NMF) method for text mining introduces a technique for partitional clustering that identifies semantic fea-
39 tures in a document collection and groups the documents into clusters on the basis of shared semantic fea-
40 tures. The factorization can be used to compute a low rank approximation of a large sparse matrix along
41 with preservation of natural data nonnegativity.

42 In the *vector space model* of text data, documents are encoded as n -dimensional vectors where n is the
43 number of terms in the dictionary, and each vector component reflects the importance of the corresponding
44 term with respect to the semantics of a document (Berry, Drmač, & Jessup, 1999). A collection of docu-
45 ments can, thus, be represented as a term-by-document matrix. Since each vector component is given a po-
46 sitive value (or weight) if the corresponding term is present in the document and a null or zero value
47 otherwise, the resulting term-by-document matrix is always nonnegative. This inherent data nonnegativity
48 is preserved by the NMF method as a result of constraints (placed on the factorization) that produce non-
49 negative lower rank factors that can be interpreted as semantic features or patterns in the text collection.
50 The vectors or documents in the original matrix can be reconstructed by combining these semantic features,
51 and documents that have common features can be viewed as a cluster. As shown by Xu, Liu, and Gong
52 (2003), NMF outperforms traditional vector space approaches to information retrieval (such as latent
53 semantic indexing) for document clustering on a few topic detection benchmark collections.

54 2. Related work

55 Nonnegative matrix factorization differs from other rank reduction methods for vector space models in
56 text mining, e.g., principal component analysis (PCA) or vector quantization (VQ), due to use of con-
57 straints that produce nonnegative basis vectors, which make possible the concept of a *parts-based represen-*
58 *tation* (Lee & Seung, 2001). Lee and Seung first introduced the notion of parts-based representations for
59 problems in image analysis or text mining that occupy nonnegative subspaces in a vector-space model.
60 Techniques like PCA and VQ also generate basis vectors—various additive and subtractive combinations
61 of which can be used to reconstruct the original space. But the basis vectors for PCA and VQ contain neg-
62 ative entries and cannot be directly related to the the original vector space to derive meaningful interpre-
63 tations. In the case of NMF, the basis vectors contain no negative entries—this allows only additive
64 combinations of the vectors to reproduce the original. So the perception of the whole, be it an image or
65 a document in a collection, becomes a combination of its parts represented by these basis vectors. In text
66 mining, the vectors represent or identify semantic features, i.e., a set of words denoting a particular concept
67 or topic. If a document is viewed as a combination of basis vectors, then it can be categorized as belonging
68 to the topic represented by its principal vector. Thus, NMF can be used to organize text collections into
69 partitional structures or clusters directly derived from the nonnegative factors.

70 Recently Xu et al. (2003) have demonstrated that NMF outperforms methods such as singular value
 71 decomposition and is comparable to graph partitioning methods that are widely used in clustering text doc-
 72 uments. The tests were conducted on two different datasets: the Reuters data corpus¹ and TDT2 corpus²,
 73 both considered benchmark collections for topic detection. These two data corpora are also used in this
 74 study to observe the results of using nonnegative factorization for text mining or document clustering.
 75 The algorithm used to derive the factorization introduces a new parameter to control the number of basis
 76 vectors used to reconstruct the document vectors, thereby providing a mechanism to balance the tradeoff
 77 between accuracy and computational cost (including storage).

78 3. Algorithm

79 With the standard vector space model, a set of documents S can be expressed as a $m \times n$ matrix V , where
 80 m is the number of terms in the dictionary and n is the number of documents in S . Each column V_j of V is
 81 an encoding of a document in S and each entry v_{ij} of vector V_j is the significance of term i with respect to the
 82 semantics of V_j , where i ranges across the terms in the dictionary. The NMF problem is defined as finding a
 83 low rank approximation of V in terms of some metric (e.g., the norm) by factoring V into the product (WH)
 84 of two reduced-dimensional matrices W and H . Each column of W is a basis vector, i.e., it contains an
 85 encoding of a semantic space or concept from V and each column of H contains an encoding of the linear
 86 combination of the basis vectors that approximates the corresponding column of V . Dimensions of W and
 87 H are $m \times k$ and $k \times n$ respectively, where k is the reduced rank or selected number of topics. Usually k is
 88 chosen to be much smaller than n , but more accurately, $k \ll \min(m, n)$. Finding the appropriate value of k
 89 depends on the application and is also influenced by the nature of the collection itself (Guillamet & Vitria,
 90 2002).

91 Common approaches to NMF obtain an approximation of V by computing a (W, H) pair to minimize
 92 the Frobenius norm of the difference $V - WH$. The problem can be cast in the following way (Pauca, Shah-
 93 naz, Berry, & Plemmons, 2004)—let $V \in R^{m \times n}$ be a nonnegative matrix and $W \in R^{m \times k}$ and $H \in R^{k \times n}$ for
 94 $0 < k \ll \min(m, n)$. Then, the objective function or minimization problem can be stated as

$$\min_{W, H} \|V - WH\|_F^2, \quad (1)$$

98 with $W_{ij} > 0$ and $H_{ij} > 0$ for each i and j .

99 The matrices W and H are not unique. Usually H is initialized to zero and W to a randomly generated
 100 matrix where each $W_{ij} > 0$ and these initial estimates are improved or updated with alternating iterations of
 101 the algorithm. In the following subsections some existing NMF techniques are discussed and a new algo-
 102 rithm is proposed.

103 3.1. Multiplicative method

104 The NMF method proposed by Lee and Seung is based on multiplicative update rules of W and H . This
 105 scheme is referred to as the multiplicative method (MM).

106 MM Algorithm

- 107 (1) Initialize W and H with nonnegative values.
 108 (2) Iterate for each c, j , and i until convergence or after l iterations:

¹ Reuters-21578 at <http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html>.

² <http://www.lcd.upenn.edu>.

$$\begin{aligned}
 109 \quad (a) \quad & H_{cj} \leftarrow H_{cj} \frac{(W^T V)_{cj}}{(W^T W H)_{cj} + \epsilon} \\
 111 \\
 112 \quad (b) \quad & W_{ic} \leftarrow W_{ic} \frac{(V H^T)_{ic}}{(W H H^T)_{ic} + \epsilon}
 \end{aligned}$$

113 In steps 2(a) and (b), ϵ , a small positive parameter equal to 10^{-9} , is added to avoid division by zero. As
 114 observed from the MM Algorithm, W and H remain nonnegative during the updates. Simultaneous updat-
 115 ing of W and H generally yield better results than updating each matrix factor fully before the other. In the
 116 algorithm, the columns of W or the basis vectors are normalized at each iteration; in case of W , the opti-
 117 mization is performed on a unit hypersphere with the columns of W effectively being mapped to the surface
 118 of the hypersphere by repeated normalization (Pauca et al., 2004).

119 The computational complexity of MM can be shown to be $O(kmn)$ operations (for a rank- k approxima-
 120 tion) per iteration (Pauca et al., 2004). Once the term-by-document matrix V has been factored into W and
 121 H , if new data needs to be added, then the data can be a direct addition to W with a minor modification to
 122 H if k is not fixed. In case of a fixed k , the new data can be integrated by further iterations with W and H as
 123 the initial approximations. It is shown by Lee and Seung (2001) that under the MM-update rules, the objec-
 124 tive function (1) is monotonically nonincreasing and becomes constant if and only if W and H are at a sta-
 125 tionary point. This multiplicative method is related to expectation-maximization approaches used in image
 126 restoration, e.g. (Prasad, Torgersen, Pauca, Plemmons, & van der Gracht, 2003), and can be classified as a
 127 diagonally-scaled gradient descent method (Guillamet & Vitria, 2002).

128 3.2. Sparse encoding

129 A new nonnegative sparse encoding scheme, based on the study of neural networks has been suggested
 130 by Hoyer (2002). This scheme is applicable to the decomposition of datasets into independent feature sub-
 131 spaces by Hyvärinen and Hoyer (2000). The method proposed by Hoyer (2002, 2004) has an important fea-
 132 ture that enforces a statistical sparsity of the H matrix. As the sparsity of H increases, the basis vectors
 133 become more localized, i.e., the parts-based representation of the data in W becomes more and more en-
 134 hanced. Mu, Plemmons, and Santago (2003) have put forth a regularization approach that achieves the
 135 same objective of enforcing statistical sparsity of H by using a point-count regularization scheme that
 136 penalizes the number of nonzero entries rather than the sum of entries $\sum_{ij} H_{ij}$ in H .

137 3.3. A hybrid method

138 The NMF algorithm used in this study (Pauca et al., 2004) is a hybrid method that combines some of the
 139 better features of the methods discussed in the previous sections. In this approach, the multiplicative meth-
 140 od, which is basically a version of the gradient descent optimization scheme, is used at each iterative step to
 141 approximate the basis vector matrix W . H is calculated using a constrained least squares (CLS) model as
 142 the metric. It serves to penalize the nonsmoothness and nonsparsity of H ; as a result of this penalization,
 143 the basis vectors or topics in W become more localized, thereby reducing the number of vectors needed to
 144 represent each document. The method for approximating H is similar to the methods described in Hoyer
 145 (2002) and Mu et al. (2003) and related to the least squares Tikhonov regularization technique commonly
 146 used in image restoration (Prasad et al., 2003). This hybrid algorithm is denoted by GD-CLS (gradient des-
 147 cent with constrained least squares) in (Pauca et al., 2004). Although there are several alternative algo-
 148 rithms for nonnegative matrix factorization (Lee & Seung, 2001; Liu & Yi, 2003), the choice of a *best*
 149 algorithm is generally considered to be problem-dependent. The proposed GD-CLS algorithm has per-
 150 formed quite well on the topic detection experiments discussed in Section 5.

151 GD-CLS Algorithm

152 (1) Initialize W and H with nonnegative values, and scale the columns of W to unit norm.153 (2) Iterate until convergence or after l iterations:154 (a) $W_{ic} \leftarrow W_{ic} \frac{(VH^T)_{ic}}{(WHH^T)_{ic} + \epsilon}$, for c and i [$\epsilon = 10^{-9}$]155 (b) Rescale the columns of W to unit norm

156 (c) Solve the constrained least squares problem:

$$\min_{H_j} \{ \|V_j - WH_j\|_2^2 + \lambda \|H_j\|_2^2 \},$$

159 where the subscript j denotes the j th column, for $j = 1, \dots, m$. Any negative values in H_j are set to zero. The parameter λ is a regularization value that is used to balance the reduction of the metric

$$\|V_j - WH_j\|_2^2$$

164 with the enforcement of smoothness and sparsity in H .166 **4. Software implementation**

167 Two software packages, namely GTP and LAPACK, were used in the C-based implementation of GD-
 168 CLS algorithm. The General Text Parser (GTP) is a software environment developed at the University of
 169 Tennessee by Giles, Wo, and Berry (2003). One of the functions of GTP is to parse text documents and
 170 construct a sparse matrix data structure, i.e., a term-by-document matrix that defines the relationship be-
 171 tween the documents and the parsed terms (Mironova, 2003). The GTP software can be used to parse
 172 single files or entire directories and is fitted with the capability to process both raw text and HTML files.
 173 The user can also integrate external filters into the software to process other forms of tagged data. Cur-
 174 rently there are two versions of the software available—one in C++ and another in Java—both of which
 175 are designed to facilitate users with all ranges of expertise. For this study, the C++ version of GTP was
 176 used.

177 LAPACK has various routines, which can be individually downloaded from the LAPACK website³, for
 178 solving different types of linear equations. For the C version of NMF, the *dposv* software routine of LA-
 179 PACK is used to derive solutions (in double precision) to linear systems of the form $AX = B$, where A is a
 180 symmetric positive definite matrix.

181 **5. Experiments**

182 Originally written in MATLAB (see Pauca et al. (2004) and Fig. 1) the proposed NMF algorithm or
 183 GD-CLS has been converted to C in this study for scalability. Performance evaluations are conducted using
 184 two different datasets—the Reuters Document Corpus and TDT2. This section describes the methodology
 185 used for evaluation, while the actual results⁴ are discussed in Section 6.

³ <http://www.netlib.org/lapack>.

⁴ All results are collected on a Sun Microsystems SunBlade 1000 workstation with 500 MHz UltraSPARC-IIe processor, 256 KB L2 cache, 512 MB DRAM and 20 GB internal disk.

```

[W, H] = gdcls(V, k, maxiter, lambda, options)
myeps = 10^-9;
if strcmp(options, 'nonneg')
    neg = 1;
else
    neg = 0;
end
[m,n] = size(V);
W = rand(m, k);
H = zeros(k, n);
for j = 1 : maxiter,
    A = W' * W + lambda * eye(k);
    for i = 1 : n
        b = W' * V(:,i);
        H(:,i) = A \ b;
    end
    if neg == 1
        H = H .* (H > 0);
    end
    W = W .* (V * H') ./ (W * (H * H') + myeps);
end

```

Fig. 1. MATLAB implementation of GS-CLS Algorithm.

186 5.1. Reuters

187 The Reuters data corpus ⁵, contains 21 578 documents and 135 topics or document clusters created man-
 188 ually and each document in the corpus is been assigned one or more topics or category labels based on its
 189 content. The manually created cluster sizes, i.e., the number of documents assigned to the topics, range any-
 190 where from less than ten to nearly four thousand topics. The documents are in SGML format (see [Shahnaz,](#)
 191 [2004](#)) with meta tags denoting title, topic(s), and beginning and end of content.

192 For this experiment, documents associated with only one topic are used and topics with cluster sizes
 193 smaller than five are discarded. To achieve this, a Perl script (see [Shahnaz, 2004](#)) is used to traverse through
 194 the corpus and create an index of topics with associated cluster sizes, where a document is considered part
 195 of a cluster only if it has a single topic.

196 In order to observe the performance of the GD-CLS implementation of NMF as the complexity of the
 197 problem increases, i.e., as the number of clusters or the parameter k is incremented, seven different k values
 198 2, 4, 6, 8, 10, 15, 20 are chosen. For each k , three different document collections or subsets are generated by
 199 the filter using different topic files, which result in creation of three term-by-document (sparse) matrices for
 200 each k . As the predominant number of elements of these matrices are zero, the Harwell-Boeing (HB) col-
 201 umn-compressed sparse matrix format ([Berry & Browne, 1999](#)) is used to access all nonzero elements. After
 202 the HB matrices are generated, the NMF clustering algorithm is performed on all 21 matrices ($7k$ values \times
 203 3 document subsets each) to produce the W and H factors for each HB matrix. For any given HB matrix V ,
 204 with k topics and n documents, matrix W has k columns or basis vectors that represent the k clusters, while
 205 matrix H has n columns that represent the n documents. A column vector in H has k components, each of
 206 which denotes the contribution of the corresponding basis vector to that column or document. The classi-
 207 fication or clustering of documents is then performed based on the index of the highest value of k for each
 208 document. So, for document i ($i = 1, \dots, n$), if the maximum value is the j th entry ($j = 1, \dots, k$), document i is

⁵ <http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html>.

209 assigned to cluster j . After the documents are clustered into k topics, the NMF generated k clusters are com-
 210 pared to the original k clusters using a mapping function. The mapping is performed using a Perl script that
 211 assigns the original cluster labels to the NMF clusters based on a similarity measure. The example on the
 212 next page provides an explanation of the mapping process for $k = 2$.

213 Once the relabeling is accomplished, the accuracy of the classification or clustering is assessed using the
 214 metric AC (Xu et al., 2003) defined by

$$AC = \sum_{i=1}^n \delta(d_i)/n,$$

217 where $\delta(d_i)$ is set to 1 if d_i has the same topic label for both NMF and the original classification, and set to 0
 218 otherwise, and n is the number of documents in the collection. So, for our two-cluster example

$$AC = \{\delta(d_1) + \delta(d_2) + \delta(d_3) + \delta(d_4) + \delta(d_5)\}/5 \\ = \{1 + 1 + 1 + 1 + 0\}/5 = 4/5 = 0.8.$$

221 In the GD-CLS implementation of NMF, the contribution of the λ parameter with which the sparsity of
 222 H is controlled is also of interest. Hence, for each k , results for three different λ values (0.1, 0.01, 0.001) are
 223 calculated.

224 Two-Cluster Example.

Original Topic Set $T = \{A, B\}$

Document subset $D = \{d_1, d_2, d_3, d_4, d_5\}$

Cluster_A = $\{d_2, d_3\}$, Cluster_B = $\{d_1, d_4, d_5\}$

227 Using GD-CLS on the HB matrix generated from D with topic set T yields WH , where $W \in R^{m \times 2}$ and
 228 $H \in R^{2 \times 5}$. Assuming H has the value shown in Table 1, the clustering based on the maximum column entry
 229 is

Cluster₁ = $\{d_2, d_3, d_5\}$,

Cluster₂ = $\{d_1, d_4\}$.

232 The values of the following mapping function are used to form a matrix S (Table 2), where $S_{iX} = \text{simi-}$
 233 $\text{lar}(\text{Cluster}_i, \text{Cluster}_X) = \text{number of documents in Cluster}_i \text{ that appear in Cluster}_X, i = (1,2) \text{ and}$
 234 $X = \{A, B\}$.

Table 1

The $2 \times 5H$ matrix for the two-cluster example (maximum entries are represented in boldface)

d_1	d_2	d_3	d_4	d_5
0.3	1.2	0.2	0.01	2.1
1.4	0.9	0.01	1.4	1.9

Table 2

The S matrix for the two-cluster example

	Cluster _A	Cluster _B
Cluster ₁	2	1
Cluster ₂	0	2

Table 3

Comparison between original cluster labels and GD-CLS generated labels for the two-cluster example

Document	Original label	GD-CLS label
d_1	B	B
d_2	A	A
d_3	A	A
d_4	B	B
d_5	B	A

235 Each Cluster_{*i*} is assigned the original cluster label to which it is the most similar. Cluster₁ and Cluster₂
 236 are assigned labels A and B respectively and the documents are reassigned to topics based on the new clus-
 237 tering. A comparison of the original clustering to the GD-CLS generated cluster labels is shown in Table 3.

238 5.2. TDT2

239 The second data corpus TDT2, obtained from the Language Data Consortium at The University of
 240 Pennsylvania⁶, contains transcripts from a total of six news sources⁷ in 3440 files, with each file containing
 241 several transcripts or documents. Although the corpus consists of about sixty-four thousand documents in
 242 SGML format (see Shahnaz, 2004), some fourteen thousand of these are actually assigned a topic label and
 243 the rest are not classified. Among the preclassified documents, 7919 documents are single topic documents,
 244 i.e., these documents only have a single topic or category label. The SGML markup tags for each document
 245 denote a unique document ID or identification number and the beginning and end of text content. The doc-
 246 ument-topic relationships are described in a separate file that contains a line in it for each document with a
 247 category label. A line corresponding to a particular document consists of the document ID, topic label, and
 248 the name of the file containing that document.

249 In order to make the document collection from this corpus comparable to the Reuters dataset, some pre-
 250 processing with the use of Perl scripts is applied to the SGML files. First, the file containing the document-
 251 topic relationships is parsed and a *topic file* or a file containing a list of 73 topics that have cluster sizes of at
 252 least five documents is created. Here also, as with the Reuters collection, documents containing multiple
 253 topic labels are not deemed relevant. Since the entire document corpus consists of 64000 documents and
 254 only 7919 are relevant to the experiments, another preprocessing step is taken to reduce the runtime of
 255 GTP by traversing the entire collection once and writing the relevant documents to a single file. For all sub-
 256 sequent testing, only this file is then used in order to avoid traversing thousands of irrelevant documents for
 257 each test run. Once the topic file and the reduced set of 7919 documents are at hand, several subsets are
 258 created to monitor the decline of accuracy for the NMF algorithm as complexity or the k values increase.
 259 As before, 7 different k values (2, 4, 6, 8, 10, 15, 20) are chosen with 10 different topic sets or document
 260 subsets each. After application of the GD-CLS algorithm and the accuracy metric, this selection of datasets
 261 yielded some of the results presented in the next section.

262 6. Observations

263 The results from TDT2 and Reuters data corpora bring to attention trends such as the decline in accu-
 264 racy in relation to the increase in complexity or the value of k . Results from both document collections indi-

⁶ <http://www.lcd.upenn.edu>.

⁷ ABC, CNN, VOA, NYT, PRI, and APW.

Table 4

Performance of the GD-CLS algorithm on the Reuters collection on a Sun Microsystems SunBlade 1000 workstation (500 MHz). The number of topics is denoted by k , the regularization parameter to control the sparsity of the H matrix factor is λ and AC denotes the accuracy measure defined in Section 5.1

k	λ	AC	CPU time (sec)
2	0.100	0.962256	2.63
2	0.010	0.963440	2.76
2	0.001	0.962262	3.19
4	0.100	0.758630	3.86
4	0.010	0.774503	4.43
4	0.001	0.777460	5.51
6	0.100	0.716229	6.51
6	0.010	0.722549	8.01
6	0.001	0.726186	10.54
8	0.100	0.572499	9.73
8	0.010	0.555926	12.79
8	0.001	0.560444	18.39
10	0.100	0.657349	30.65
10	0.010	0.673601	36.79
10	0.001	0.666243	47.75
15	0.100	0.609148	56.53
15	0.010	0.613033	74.89
15	0.001	0.618249	104.18
20	0.100	0.545806	57.26
20	0.010	0.567711	87.77
20	0.001	0.571387	122.13

cate that as more and more topics or document clusters are added to the dataset being clustered by GD-CLS, the accuracy of the clustering decreases. For the Reuters collection, in case of $k = 2$, i.e., when dealing with only two topics, the algorithm performs with above 99% accuracy, but in case of $k = 20$, the accuracy drops down to just above 54% (Table 4). However, in case of TDT2, the drop in accuracy is much less precipitous than for Reuters (Table 5). For TDT2, for $k = 20$, accuracy is just above 80%, which seems like a significant improvement from 54% for Reuters. This disparity can be attributed to the differences in content of the two collections. Documents in the Reuters collection are categorized under broad topics (such as “earn,” “interest,” “cocoa,” “potato,” etc., listed in Shahnaz (2004)), while for TDT2, the topic labels are much more specific (e.g., “The Asian economic crisis,” “Tornado in Florida,” and “Oprah lawsuit”). The very specificity of the topics in the TDT2 guarantees a heterogeneity in the document collection that is not present in the Reuters collection. In the case of Reuters, while “potato” and “zinc” may constitute very distinct clusters, “interest” and “money-fixes” do not. In fact, as noted by Xu et al. (2003), there is a degree of overlapping of content across topics in the Reuters collection that contributes to the much more rapid decline of accuracy in case of Reuters than it does for TDT2.

Another notable trend that also points to the sensitivity of the GD-CLS algorithm (for NMF) to the contents of the document collections is the differences in accuracy for the different λ values. In case of TDT2, the different λ values for each k do not affect the performance by any noticeable amount. But for Reuters, the drop in accuracy for increasing values of the λ parameter suggests that text collections that are somewhat homogeneous in content, are more sensitive to the changes of the λ parameter (or the sparsity of the H matrix). The primary reason for using a larger λ value (or an increase in the sparsity of H) is to

Table 5

Performance of the GD-CLS algorithm on the TDT2 collection on a Sun Microsystems SunBlade 1000 workstation (500 MHz). The number of topics is denoted by k , the regularization parameter to control the sparsity of the H matrix factor is λ and AC denotes the accuracy measure defined in Section 5.1

k	λ	AC	CPU time (sec)
2	0.100	0.993629	2.93
2	0.010	0.993629	2.94
2	0.001	0.978329	3.00
4	0.100	0.906264	9.42
4	0.010	0.908873	9.48
4	0.001	0.925784	10.04
6	0.100	0.878919	23.38
6	0.010	0.858782	23.60
6	0.001	0.860544	25.81
8	0.100	0.858497	46.86
8	0.010	0.859123	47.42
8	0.001	0.853479	52.48
10	0.100	0.840443	97.39
10	0.010	0.836955	98.34
10	0.001	0.847155	110.26
15	0.100	0.869069	135.66
15	0.010	0.872499	140.08
15	0.001	0.870932	172.06
20	0.100	0.832097	303.54
20	0.010	0.835903	315.64
20	0.001	0.840977	405.16

Table 6

Elapsed CPU time for the GD-CLS algorithm in the case of $k = 15$ topics and different choices of the regularization parameter λ . All experiments were run on a Sun Microsystems SunBlade 1000 workstation (500 MHz)

λ	Reuters	TDT2
0.1	56.5433	135.6620
0.01	66.0033	140.0820
0.001	93.3900	172.0670

285 achieve faster computation times. Inspection of the results from Tables 4 and 5 suggests that that is indeed
286 the case, especially in higher complexity problems (Table 6).

287 Following (Karvanen & Cichocki, 2003), we have measured the increase in sparsity of $H = [h_{ij}]$ using the
288 ℓ_p matrix norm (for $p < 1$) which is defined by

$$\|H\|_p = \left[\sum_i \sum_j (h_{ij})^p \right]^{1/p}.$$

291 Figs. 2 and 3 illustrate the consistent reduction (y -axis) in $\|H\|_p/\|H\|_1$ for $p = 0.5$ and $p = 0.1$ for selected
292 $k = 10$ and $k = 20$ document subsets of the Reuters collection as the regularization parameter λ (x -axis) in-
293 creases from 10^{-6} to 1. Recall that k is the number of topics extracted for each subset and the number of
294 feature vectors (column dimension of matrix W) by the GD-CLS algorithm.

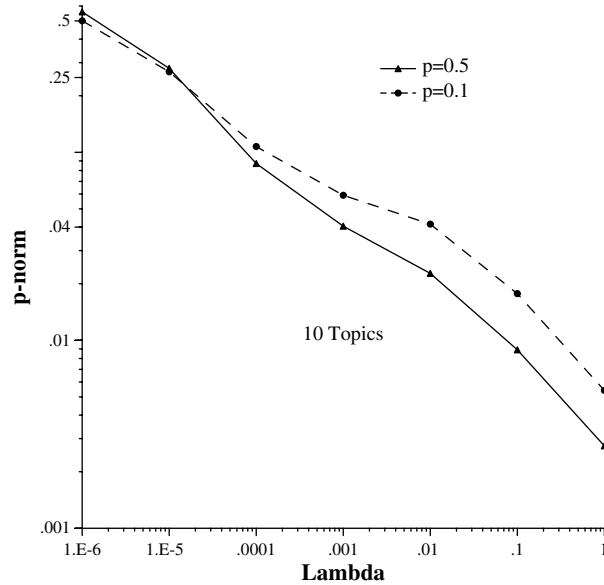


Fig. 2. Reduction in $\|H\|_p/\|H\|_1$ (p -norm) for $p = 0.5, 0.1$ as the regularization parameter λ (Lambda) is increased for the nonnegative matrix factorization $A = WH$ of a $k = 10$ topic term-by-document matrix from the Reuters collection.

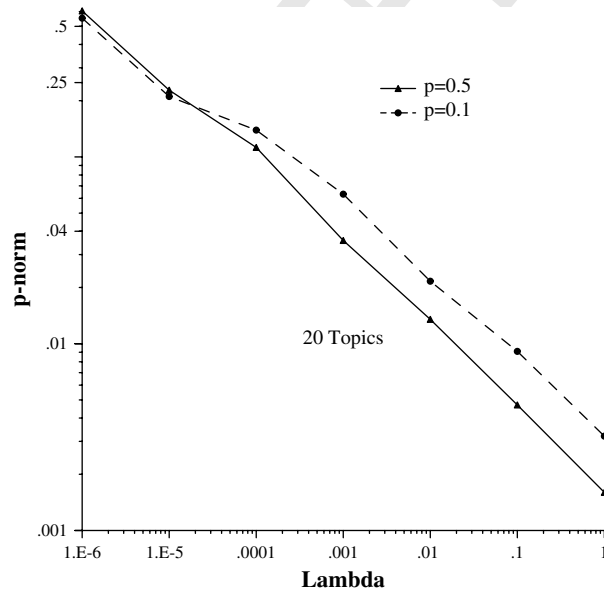


Fig. 3. Reduction in $\|H\|_p/\|H\|_1$ (p -norm) for $p = 0.5, 0.1$ as the regularization parameter λ (Lambda) is increased for the nonnegative matrix factorization $A = WH$ of a $k = 20$ topic term-by-document matrix from the Reuters collection.

295 It can be inferred from Table 6 that an increase in the sparsity of H results in a significant increase in
 296 computational speed and this holds for both TDT2 and Reuters. As for accuracy, the λ values do affect
 297 performance for Reuters but not for TDT2. However, when compared to the gain in computational time,
 298 the 2–3% decrease in accuracy can be considered a very reasonable tradeoff.

Table 7

Output clusters generated by the GD-CLS algorithm for both similar and disparate (original) clusters of documents from the Reuters and TDT2 collections

Corpus	Dataset	Cluster	Original cluster sizes	GD-CLS generated cluster sizes
Reuters	dataset ₁	cluster ₁	2125	1690
		cluster ₂	45	480
	dataset ₂	cluster ₁	114	112
		cluster ₂	99	101
TDT2	dataset ₁	cluster ₁	1476	1231
		cluster ₂	31	276
	dataset ₂	cluster ₁	110	109
		cluster ₂	120	121

299 An aspect of GD-CLS that cannot be directly observed from the result tables is the change in perfor-
 300 mance of the factorization with regards to disparate cluster sizes. When creating document subsets for each
 301 value of k from the preclassified clusters of the Reuters or TDT2 corpus, attention is given to keep the clus-
 302 ter sizes within a reasonable bound of one another. This constraint, which is not imposed by Xu et al. in
 303 (Xu et al., 2003), is enforced due to results obtained from experiments similar to those described in Table 7.

304 The imbalance in the cluster sizes in dataset₁ has a definite effect on the performance of GD-CLS regard-
 305 less of the document corpus being used. In case of the original clusters from dataset₁, the ratio of cluster₁ to
 306 cluster₂ is approximately 48:1, while the clusters produced by GD-CLS have a ratio of 3:1. This implies
 307 GD-CLS performs much better on datasets that have balanced cluster sizes, such as dataset₂, where clus-
 308 tering is performed with almost 100% accuracy.

309 7. Conclusions and future work

310 Clustering of documents in a database according to semantic features inherent to the data is a challeng-
 311 ing problem in text data mining. In this paper we present a new approach for *unsupervised* identification of
 312 semantics topics and associated document clustering in a heterogeneous text collection. This approach is
 313 based on the use of nonnegative matrix factorization algorithms for the factorization of term–document
 314 matrices. Additional statistical sparsity constraints are imposed. A new algorithm, GD-CLS, is presented
 315 for factoring a term–document matrix into a matrix W of basis vectors denoting semantic features and a
 316 matrix H of mixing coefficients specifying document clustering. The sparsity constraint is enforced via a
 317 parameter λ used for controlling document clustering as well as to balance computational cost and accu-
 318 racy, as illustrated in our results. Other variations of GD-CLS have been explored by the authors in order
 319 to consider both accuracy and efficiency. For example, nonnegative least squares methods can be used to
 320 compute W and H at each iteration in order to speed up convergence, at the cost of a significant increase in
 321 computation per iteration. On the other hand, gradient descend type methods can be used for updating W
 322 and H efficiently at the cost of slow convergence. The GD-CLS algorithm attempts to strike a balance be-
 323 tween both of these extremes.

324 Updating semantic features and document clusters is often needed as documents are added to a text col-
 325 lection. In its current stage, the GD-CLS algorithm is not equipped to handle updating in an efficient man-
 326 ner. Once the document collection has been clustered, adding a small number of documents to the
 327 collection can be achieved by comparing each of the new documents (represented by a vector) to the basis
 328 vectors and associating the new document to the basis vector or topic to which it is most similar. But this
 329 updating technique is not scalable and would produce poor results if used to add a large number of doc-

330 uments that cannot be associated with any of the basis vectors. Techniques for efficiently updating features
331 and clusters as documents are added to a text collection are currently under investigation.

332 Although the primary function of NMF would be for classification as opposed to query-based informa-
333 tion retrieval, the resulting clusters can be used to provide retrieval capabilities. Much in the style of limited
334 updating discussed above, a user query can be represented by a term vector, which is then used to compute
335 a similarity measure (e.g., cosine measurement) between the query and the basis vectors. The basis vector or
336 topic that yields the highest value is deemed the most relevant and documents belonging to that topic are
337 provided to the user.

338 In general, NMF has mostly been applied to image analysis and text mining. Another field that could
339 benefit from this technique is bioinformatics. Problems such as identifying motifs or significant features
340 in protein sequences (partial strings of DNAs) are natural candidates for application of NMF. In such
341 problems, protein sequences can be viewed as analogous to text documents and the basis vectors or topics
342 to motifs that control gene expression (Stuart & Berry, 2003). We are currently investigating the effect of
343 enforcing adequate constraints, such as statistical sparsity, to these types of problems.

344 Acknowledgements

345 The authors would like to thank Murray Browne for his technical assistance with the production of the
346 original manuscript and the anonymous referees for their comments and suggestions for the subsequent
347 revision.

348 References

- 349 Berry, M., & Browne, M. (1999). *Understanding search engines: mathematical modeling and text retrieval*. Philadelphia, PA: SIAM.
- 350 Berry, M., Drmač, Z., & Jessup, E. (1999). Matrices, vector spaces, and information retrieval. *SIAM Review*, 41, 335–362.
- 351 Dunham, M. (2003). *Data mining: introductory and advanced topics*. Upper Saddle River, NJ: Prentice Hall.
- 352 Giles, J., Wo, L., & Berry, M. (2003). GTP (general text parser) software for text mining. In H. Bozdogan (Ed.). *Software for text*
353 *mining, in statistical data mining and knowledge discovery* (pp. 455–471). Boca Raton, FL: CRC Press.
- 354 Guillaumet, D., & Vitria, J. (2002). Determining a suitable metric when using non-negative matrix factorization. In *Sixteenth*
355 *international conference on pattern recognition (ICPR'02)*, Vol. 2. Quebec City, QC, Canada.
- 356 Hoyer, P. (2002). Non-negative Sparse Coding. In *Proceedings of the IEEE workshop on neural networks for signal processing*.
357 Martigny, Switzerland.
- 358 Hoyer, P. (2004). Non-negative matrix factorization with sparseness constraints. Tech. rep., Helsinki Institute for Information
359 Technology, University of Helsinki, Finland, preprint.
- 360 Hyvärinen, A., & Hoyer, P. (2000). Emergence of phase and shift invariant features by decomposition of natural images into
361 independent feature subspaces. *Neural Computation*, 12, 1705–1720.
- 362 Karvanen, J., & Cichocki, A. (2003). Measuring sparseness of noisy signals. In *Proceedings of the fourth international symposium on*
363 *independent component analysis and blind signal separation (ICA2003)*. Nara, Japan.
- 364 Lee, D., & Seung, H. (2001). Algorithms for non-negative matrix factorization. *Advances in Neural Information Processing Systems*, 13,
365 556–562.
- 366 Liu, W., & Yi, J. (2003). Existing and New algorithms for nonnegative matrix factorization. Tech. rep., Department of Computer
367 Sciences, University of Texas at Austin.
- 368 Mironova, S. (2003). Integrating network storage into information retrieval applications. Master's thesis, Department of Computer
369 Science, University of Tennessee, Knoxville, TN.
- 370 Mu, Z., Plemmons, R., & Santago, P. (2003). Iterative ultrasonic signal and image deconvolution for estimating the complex medium
371 response. In *IEEE transactions on ultrasonics and frequency control*. IEEE, submitted for publication.
- 372 Pauca, V., Shahnaz, F., Berry, M., & Plemmons, R. (2004). Text mining using nonnegative matrix factorizations. In *Proceedings of the*
373 *fourth SIAM international conference on data mining*, April 22–24. SIAM, Lake Buena Vista, FL.
- 374 Prasad, S., Torgersen, T., Pauca, V., Plemmons, R., & van der Gracht, J. (2003). Restoring images with space variant blur via pupil
375 phase engineering. Optics in info. systems, special issue on Comp. Imaging, SPIE Int. Tech. Group Newsletter, Vol. 14 (2), pp. 4–5.

- 376 Shahnaz, F. (2004). Clustering method based on nonnegative matrix factorization for text mining. Master's thesis, Department of
377 Computer Science, University of Tennessee, Knoxville, TN.
- 378 Stuart, G., & Berry, M. (2003). Comprehensive whole genome bacterial phylogeny using correlated peptide motifs defined in a high
379 dimensional vector space. *Journal of Bioinformatics and Computational Biology*, 1(3), 475–493.
- 380 Xu, W., Liu, X., & Gong, Y. (2003). Document-clustering based on non-negative matrix factorization. In *Proceedings of SIGIR'03*,
381 July 28–August 1, Toronto, CA, pp. 267–273.
- 382

UNCORRECTED PROOF