Pattern Analysis and Application manuscript No.
(will be inserted by the editor)

Pairwise-adaptive dissimilarity measure for document clustering

Joris D'hondt, Joris Vertommen, Paul-Armand Verhaegen, Dirk Cattrysse, Joost R. Duflou

Centrum voor Industrieel Beleid
Katholieke Universiteit Leuven
Celestijnenlaan 300A bus 2422
3001 Heverlee Belgium

Received: date / Revised version: date

Abstract During the last decades the amount of digitally available information has drastically increased. Manually processing the necessary information required by various types of applications has become infeasible. There is a need to support these applications by introducing a certain level of automation in the information handling process. One popular approach in text based systems is the application of cluster algorithms on a collection of documents to discover one or more hidden structures in this collection. In the process of clustering, proximity measures play a central role. Many clustering algorithms start from a square matrix, describing the proximity of all items in the dataset towards each other. Based on the knowledge of proximity, several coherent groups of items can be identified. This paper introduces a novel pairwise-adaptive dissimilarity measure that improves clustering quality and speed, when applied in document environments. Its performance is validated on several datasets.

Key words Clustering Distance Measure cluster quality

1 Introduction

Clustering already has a long history as a mental process. Mankind always had the tendency to cluster items to obtain a categorization. Early human civilizations e.g. tried to differentiate food on eatable and non-eatable food. Understanding the human language also relies on clustering: each noun in a language is essentially a label used to describe a class of objects that share several properties. The common goal of all clustering techniques is to discover an underlying structure in the investigated data collection. In modern times the advent of the computer drastically increased the amount of electronic information, such as websites, e-mail and other documents. To keep a grip on this vast collection of unstructured information, automatic ordering systems are desirable. Unsupervised clustering can form the basis of such an ordering system. This is a technique to partition data in which the assessment of the partitions is done without prior knowledge of the underlying structure of the data.

Document Clustering applied on vast document collections can offer benefits to an application in various ways. It can contribute directly as a functionality on its own, e.g. providing a topical overview of a dataset. It can also have an indirect effect by speeding up other functionalities, e.g. query matching to support an advanced text search engine.

This paper introduces a new dissimilarity measure that can be applied in clustering algorithms. The next section discusses some basic principles, as well as related work on proximity measures. Section **??** describes into the details of the proposed pairwise-adaptive dissimilarity measure. Section **??** provides an experimental validation of the developed algorithm. Finally, Section **??** offers conclusions based on the performed validation.

2 Related work

This section covers some basic issues when dealing with clustering in text environments. Furthermore, an in-depth analysis of proximity measurement is given, as this aspect of clustering forms the core of the research presented in this paper.

2.1 Document clustering

A good overview of different clustering algorithms is given in [**?**]. There are some aspects however, that almost all clustering algorithms share:

- **Data representation**: the items in the, to be clustered, dataset are represented in a uniform way. The standard approach in document clustering is to represent items using the Vector Space Model [**?**].

Send offprint requests to:

- **Proximity measurement**: the proximity between each pair of items in the dataset is measured. Proximity can be defined in function of similarity or dissimilarity. Popular measures are based on euclidean or cosine distances [**?**,**?**]. The result of the proximity measurement is a symmetric proximity matrix containing all data.
- **Linkage rules**: based on the proximity between items, the items are linked together into clusters. Clusters can be formed in a hierarchical manner, grouping items together starting the two closest, but they can also be initiated randomly.

The Vector Space Model (VSM) was used to represent dataset items in the research described here. The VSM is widely applied in the context of information retrieval and text mining. It takes a collection of documents as input, which is consequently transformed into an "index". This index can be seen as a (very large) matrix of which each column represents a document. The rows are defined by the vector space's dimensions, which are extracted from the vocabulary used in the document collection. Each unique term in the vocabulary corresponds to a dimension.

Each document (column) receives a positive weight in each dimension (row), indicating the relative importance of this dimension in presenting the content of this document. As such, the documents effectively become vectors in the defined vector space. The resulting matrix representation of the document corpus is depicted in Figure **??**.

Figure 1 Example of a term-by-document matrix with n terms describing m documents

The weights of the terms in the term-by-document matrix are assigned by a weighting scheme. There exists a multitude of weighting schemes [**?**], but all of them have in common that they give a weight of 0 to dimensions that do not occur in a specific document. The research in this paper will use the TF-IDF weighting scheme [**?**].

Because only a few terms appear in all documents, the term-by-document matrix contains a lot of zeros, i.e. it is very sparse. This property of the term-by-document matrix can be exploited to generate a presentation of the matrix that uses little memory. This is often necessary because the vocabulary that is used in a typical set of documents can easily contain thousands of terms. The high dimensionality of vector spaces that are based on text documents has some important consequences. First of all, it leads to what is known as the "curse of high dimensionality" [**?**]. This phenomenon states that in a space with high dimensionality, all points are equally far away from each other, making a distinction based on proximity impossible. Secondly, high dimensionality makes calculations in the vector space computationally intensive.

To counter these dimensionality-related issues, a preferable step in the first phase of the clustering process, is the reduction of the number of dimensions or features. In literature two major approaches can be identified to obtain a reduced number of dimensions [**?**]. The common objective of these techniques is to improve classification performance and/or computational efficiency [**?**].

Feature selection Feature selection is a well-explored topic in the domain of statistical pattern recognition [**?**]. In a clustering context no strict guidelines exist on how to devise this selection of features. It often occurs on an ad-hoc basis, possibly combined with a trial-error process [**?**]. Guiding heuristics can be based on measurements such as the IDF weight of dimensions, or the Zipf-curve [**?**]. By removing the most irrelevant and redundant features from the data, a reduction of noise in the feature set is obtained.

Feature extraction Feature extraction techniques compute new features from the original feature set. The application of feature extraction techniques is not limited to the text processing domain, they are often used in other domains such as image processing. The more widely used methods include:

- Principal components analysis [**?**]
- Semidefinite embedding [**?**]
- Latent semantic analysis [**?**]
- Partial least squares [**?**]

In literature, Principal Component Analysis or the associated Singular Value Decomposition is known to be the best performing linear dimension reduction technique [**?**][**?**]. The original term space is reduced to the spaces spanned by a selected number of eigenvectors. This reduction results in a loss of information but retains the largest part of variance, reducing the number of dimensions and therefor making the clustering process computationally feasible.

2.2 Proximity measures

Since the notion of proximity is essential to reach the objective of a clustering process, numerous measurements have already been proposed to quantify the proximity between two items in the same feature space. For instance, [**?**] alone lists more than 60 different similarity measures. However, not all proximity measures are applicable in each environment. A distinction is made between categorical and continuous data. In the context of document clustering, the data adheres to the second type. Several measures such as Pearson correlation [**?**], Jaccard [**?**] or Euclidean similarity [**?**] have their merits

in this context. It is commonly accepted however, that the cosine measure (see section **??**) is the best performing proximity measure in a text based environment [**?**].

2.2.1 Cosine distance measure The cosine distance between two documents a and b is defined as the cosine of the angle between the two document vectors, i.e.

$$s(x_a, x_b) = \frac{x_a \cdot x_b}{\|x_a\|_2 \cdot \|x_b\|_2} \tag{1}$$

The range of this dissimilarity measure is $[-1, 1]$. The related similarity measure is defined as $1 - s(x_a, x_b)$. The range of this similarity is $[0, 2]$.

2.2.2 Cosine dissimilarity measure variants Due to its general acceptance as one of the best performing dissimilarity measures, the cosine distance measure is the basis of several variants. This short overview does not intend to cover all existing variants, but rather indicates the importance of the cosine distance measure to the field of clustering.

The cosine distance measure with tolerance window is an attempt to use the original distance measure with the application of one or more boundaries [**?**]. This distance measure adapts the cosine distance to match peaks within a tolerance window e.g. to account for error between vectors originating from experimental and theoretical spectra in bioinformatics. If the tolerance window r is reduced to zero, the dissimilarity measure reduces to the original cosine measure.

The adjusted cosine similarity handles the difference in importance of the compared vectors [**?**]. A serious drawback of the original distance measure is the lack of considering the different ratings of the vectors. The adjusted cosine similarity offsets this drawback by incorporating the corresponding vector average in the similarity measure.

The binary cosine similarity measure is computed exactly the same way as the regular cosine similarity except that a word form receives a score of 1 when it appears in a document and 0 when it does not appear.

The binary Dice [**?**] or the similar Jaccard coefficient [**?**] equals the binary cosine value when the two compared document vectors contain the same number of non-zero entries. The binary cosine similarity penalizes less compared to the Dice coefficient when the number of non-zero score vector entries is very different for the two documents. The interpretation of the resulting values are similar to the cosine similarity measure: a value of 0 means that the two documents are entirely dissimilar while a value of 1 means the opposite. A statistical approach based upon the cosine similarity measure is the collection of whitened cosine similarities [**?**]. This last collection of similarity measures is extensively used in face pattern recognition.

3 Pairwise-adaptive dissimilarity measure

In order to decrease the degree of noise in the feature set, a novel dissimilarity measure is proposed based on a combination of the original cosine dissimilarity measure and an observation specific feature selection. The measure is calculated similarly to the original dissimilarity measure, but for each pair of observations the inner product is obtained within a customized subspace. The dimensionality K of this subspace is determined by the minimum of the K_1 most important features of the first vector and the K_2 most important features of the second vector, in formula this is $K = min(K_1, K_2)$. The subspace thus constructed has therefore limited dimensionality, in the range between $min(K_1, K_2)$ and $2 * min(K_1, K_2)$. The selection of these most relevant terms is based upon a sorting of the (absolute) values of the vector. This dissimilarity measure is formally written as

$$s(x_{K,a}, x_{K,b}) = \frac{x_{K,a} \cdot x_{K,b}}{\|x_{K,a}\|_2 \cdot \|x_{K,b}\|_2} \tag{2}$$

$x_{K,a}$ is the vector containing the K highest weighted dimensions of the original vector a.

The number of terms to use in the application of the dissimilarity measure can be determined in two manners. The first approach is the most straightforward: a fixed number of terms is used in the calculation procedure. This number is selected beforehand, and is based on the average stem density per document vector of the term-by-document matrix. This method has some drawbacks, such as the difference in actual stem density of the documents and the possible selection of stems in the unfavorable parts of the Zipf-Curve [**?**].

The second approach tries to overcome these drawbacks, and is based on a dynamic measurement of the number of terms of the compared document vectors. This is

$$K = p * min(K_1, K_2)$$

In this formula K_1 and K_2 are the number of unique terms that document 1 respectively 2 contains. The parameter p indicates the percentage of terms taken into account for the dissimilarity measure. In all conducted tests this parameter value was set in the range of 0.35 to 0.4 as a result of the best performance, as will be explained in **??**. For every dissimilarity calculation between a pair of documents, this K-value is recalculated. Two boundary conditions are applied to this dissimilarity calculation. If the density of one of the document vectors is below a predefined threshold, the resulting number of terms is assigned a predefined value. In all conducted test this parameter value was set to 50 terms. This figure was selected based upon several test results. An upper limit for K-value also can be identified. This upper limit garantees the computational advantage of the new dissimilarity measure to the original cosine dissimilarity

measure. This upper limit is further explained in section **??**.

This new variant on the cosine dissimilarity measure (similar to the original cosine dissimilarity measure) is not a distance measure according to the definition of a metric [**?**]. This definition states that a dissimilarity measure $d_{i,j}$ is termed as a distance measure if it complies with the triangle inequality $d_{i,j} + d_{i,m} \geq d_{j,m}$ for the pairs of data points (i, j), (i, m) and (j, m). The following term-by-document matrix clearly shows that the metric inequality is not necessarily valid when applying the new dissimilarity measure.

$$\begin{bmatrix} 1 & 2 & 4 \\ 2 & 2 & 3 \\ 0 & 0 & 2 \\ 0 & 1 & 1 \\ 3 & 3 & 0 \end{bmatrix}$$

Metric dissimilarity measures enable computational advantages such as lowering computational burden in neighborhood search due to the triangle inequality [**?**]. The newly proposed dissimilarity measure, like the original cosine measure, is therefor not suited to be applied in combination with metric based techniques.

4 Experimental validation

As stated in [**?**], a possible manner to compare different distance or dissimilarity measures is to study their retrieval performance in terms of precision and recall in several test environments. All datasets employed in these experiments were preprocessed in the same manner, were subjected to the same clustering method and were evaluated in the same manner. The only variation between the experiments was the dissimilarity measure used in calculating the clustering.

4.1 Datasets

The test sets constructed for these experiments were derived from three standardized datasets:

- OHSUMED [**?**]
- Reuters RCV1 corpus [**?**]
- Banksearch [**?**]

From these datasets, several subsets were constructed to fit the needs of this research, and will be described here. All subsets consist of documents which were grouped together according to the classification available for the OHSUMED, Reuters RCV1 and Banksearch datasets. Each document was verified to belong to only one of the included classification topics, in order to avoid overlap between the subjects as much as possible. The properties of these subsets are described in the appendices.

Table 1 Overview of the selected OHSUMED topics

OHSU 1	OHSU 14	OHSU 24	OHSU 33
OHSU 3	OHSU 18	OHSU 26	OHSU 35
OHSU 5	OHSU 19	OHSU 29	OHSU 36
OHSU 11	OHSU 21	OHSU 30	OHSU 37
OHSU 13	OHSU 23	OHSU 32	OHSU 38

Table 2 Overview of the selected Reuters topics

Defense	Human interest
Disasters & accidents	religion
Arts, culture & entertainment	Science & technology
Environment & natural world	Sports
Fashion	Travel & tourism
Obituaries	Weather
Crime & law enforcement	International relations
Health	Labor issues
Domestic politics	Personalities & people
War & civil war	Welfare & social services

4.1.1 Ohsumed 1 In this first test set, documents were selected from the following 20 categories listed in Table **??**.

These categories were chosen randomly, without knowledge about their actual content. From each topic, 15 documents were selected, and 19 subsets were constructed as follows:

- Subset one contains 30 documents covering 2 topics, OHSU 1 & OHSU 3
- Subset two contains 45 documents covering 3 topics, OHSU 1, OHSU 3 & OHSU 5
- ...
- Subset 19 contains 285 documents covering all topics

4.1.2 Ohsumed 2 The second test set also consists of documents from the OHSUMED dataset, but from 4 randomly chosen categories only:

- OHSU 3
- OHSU 11
- OHSU 30
- OHSU 33

From these topics, 20 subsets were constructed as follows:

- Subset 1 contains 200 documents, 50 from each of the above topics
- Subset 2 contains 400 documents, 100 from each of the above topics
- ...
- Subset 20 contains 4000 documents, 1000 from each of the above topics

4.1.3 Reuters 1 This test set consists of documents selected from the Reuters topics given in Table **??**.

From each topic, 15 documents were selected, and 19 subsets were constructed as follows:

– Subset one contains 30 documents covering 2 topics, 'Defense' and 'Disasters & accidents'
– Subset two contains 45 documents covering 3 topics
– ...
– Subset 19 contains 285 documents covering all topics

4.1.4 Reuters 2 This test set was constructed in the same manner as Ohsumed 2. This test set contains one topic documents, selected from 4 topics of the Reuters RCV1 dataset.

These four topics are:

– C13
– C24
– C31
– C42

4.1.5 Banksearch This multilevel dataset consists of the topics given in Table **??** From each of these topics 21 test

Table 3 Overview of the selected Banksearch categories

Commercial banks	Building societies
Insurance agencies	Java
C/C++	Visual Basic
Astronomy	Biology
Soccer	Motor sports
Sport	

sets were constructed. For every test set a random number of topics and documents were selected.

4.2 Experimental setup

All previously described test sets were treated identically before the clustering step. This treatment contained following steps:

– stopword removal using a list of English stopwords
– stemming using the Porter stemming algorithm [**?**]
– application of the tf-idf weighting scheme [**?**]

The goal of these experiments was to evaluate the variation in clustering results for two different dissimilarity measures, i.e. cosine and the adaptive cosine dissimilarity measure. The clustering algorithm used to compare the performance of the two distance measures in these experiments, was the hierarchical agglomerative clustering method [**?**] using average linkage [**?**].

4.3 Validation measure

For each test set all documents were selected so as not to belong to more than one topic. Therefore it is assumed that each topic is represented in the clustering result by one cluster of documents. Each cluster of documents is labeled with a specific topic as follows:

– Find the largest group of documents on the same topic within a cluster
– Label this cluster as the cluster covering that topic
– Mark this cluster as labeled, and the topic label as assigned
– Repeat with the cluster and topics that are left

Each clustering is given a score by calculating its average F-measure [**?**] which combines precision (p) and recall (r) in one value as

$$F(r,p) = \frac{2 \cdot r \cdot p}{r + p}$$

4.4 Results

The results of the clustering algorithm for each of the test sets are given by plotting the average F-measure scores assigned to each clustering result. This clustering step is calculated for the cosine dissimilarity measure and the pairwise adaptive dissimilarity measure. The related computational times are also given to indicate the performance improvement of the pairwise adaptive dissimilarity measure.

4.4.1 Clustering validation In this section the cluster validation results of all test sets are presented. In all subsequent figures the cosine distance measure is indicated with a dotted line. The pairwise adaptive dissimilarity measure is indicated with a dashed line. The results of applying the previous described clustering method in the two test environments derived from the standardized Reuters RCV1 document collection are shown in Figures **??** and **??**. For the Reuters1 test environment, only for three sets of the test environment the pairwise adaptive dissimilarity measure results in a lower F-measure scoring. All sets of the Reuters2 test environment result in an equal or higher F-measure score for the new dissimilarity measure. The average improvement in the Reuters1 and Reuters2 test environments are respectively 10,6% and 3,5%. The improvement is ranging from -4,5% to 41% for the first reuters test environment. For the second test environment the range is 0 to 8%. These relative figures are shown in Figures **??** and **??**.

Similar conclusions can be drawn from the two test environments originating from the standardized Ohsumed collection. The resulting figures in these two test environments are shown in Figures **??** and **??**. The average improvement in the Ohsumed1 and Ohsumed2 test environments are respectively 2,6% and 5,5%. For the Ohsumed1 test environment four test sets result in a lower F-measure. The improvement is ranging from -2 to 11% of the original cosine dissimilarity measure. For the Ohsumed2 test environment only two test sets result in a lower F-measure, the range is from -2,3 to 47,5%. These relative figures are shown in Figures **??** and **??**.

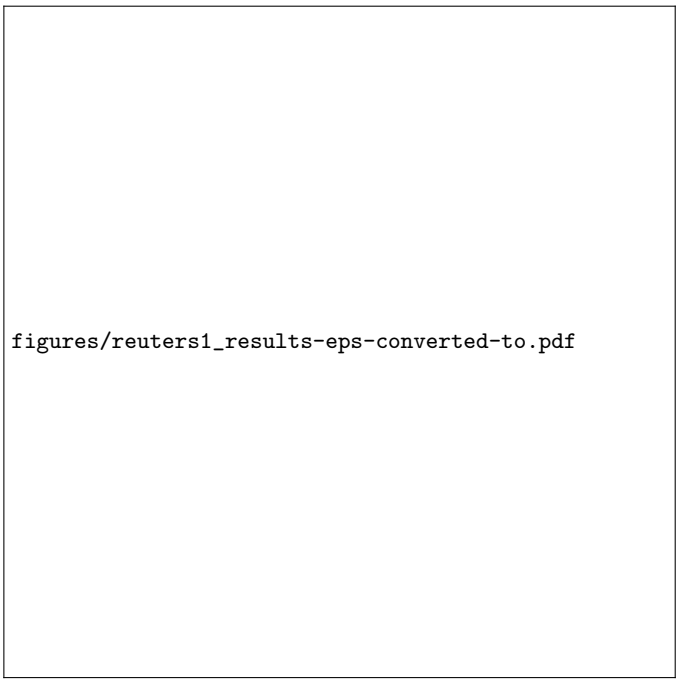

Figure 2 F-measure results of the Reuters1 test set. The original cosine measure is indicated with a dotted line. The pairwise adaptive dissimilarity measure is shown with a dashed line.

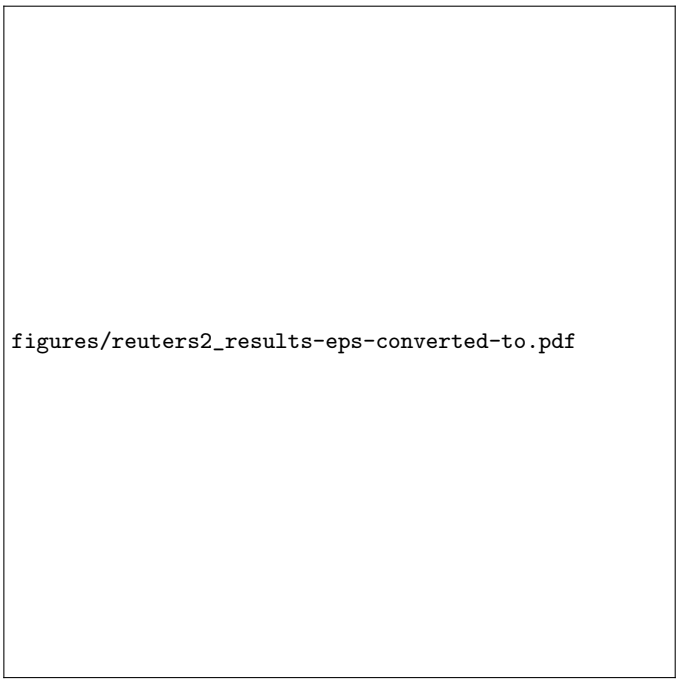

Figure 4 F-measure results of the Reuters2 test set. The original cosine measure is indicated with a dotted line. The pairwise adaptive dissimilarity measure is shown with a dashed line.

Figure 3 Relative f-measure of the Reuters1 test set. The original cosine measure is indicated with a dotted line. The pairwise adaptive dissimilarity measure is shown with a dashed line.

Figure 5 Relative f-measure of the Reuters1 test set. The original cosine measure is indicated with a dotted line. The pairwise adaptive dissimilarity measure is shown with a dashed line.

Finally the results of the Banksearch test environment are shown in Figure **??**. For three test sets the proposed dissimilarity measure results in a slightly lower F-measure value. The average improvement for this test environment is 19,5%. The improvement ranges from -9,1 to 77 %. This relative figure is shown in Figure **??**.

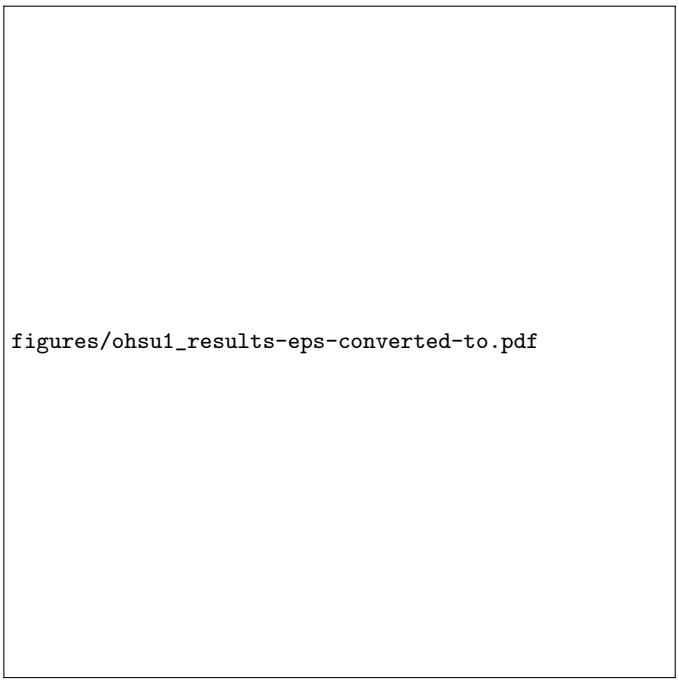

Figure 6 F-measure results of the Ohsumed1 test set. The original cosine measure is indicated with a dotted line. The pairwise adaptive dissimilarity measure is shown with a dashed line.

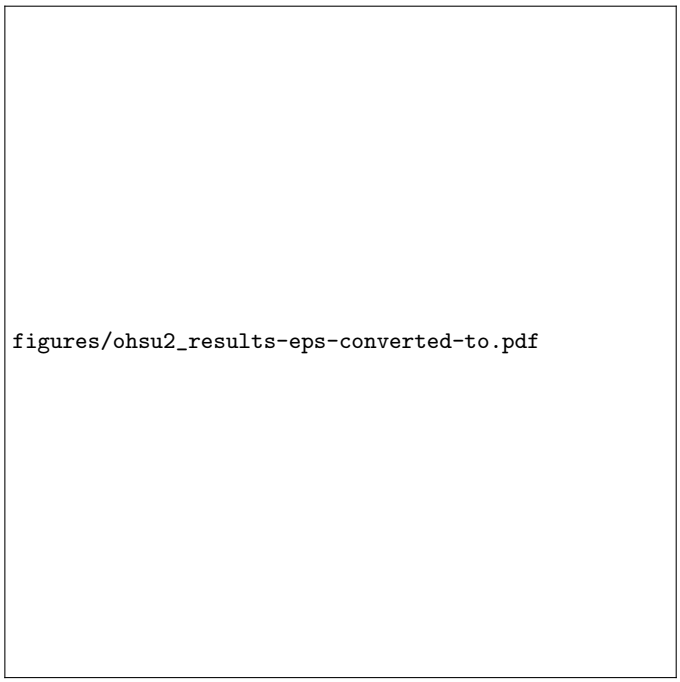

Figure 8 F-measure results of the Ohsumed2 test set. The original cosine measure is indicated with a dotted line. The pairwise adaptive dissimilarity measure is shown with a dashed line.

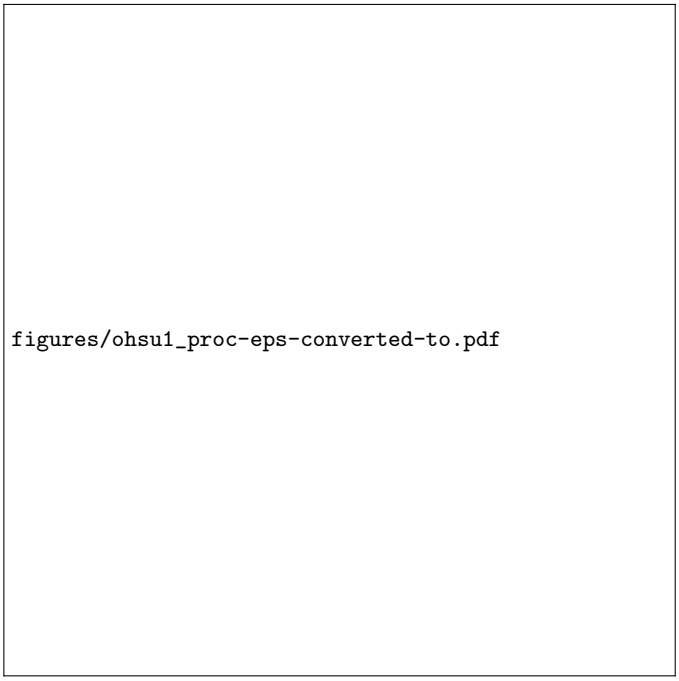

Figure 7 Relative f-measure of the Ohsumed1 test set. The original cosine measure is indicated with a dotted line. The pairwise adaptive dissimilarity measure is shown with a dashed line.

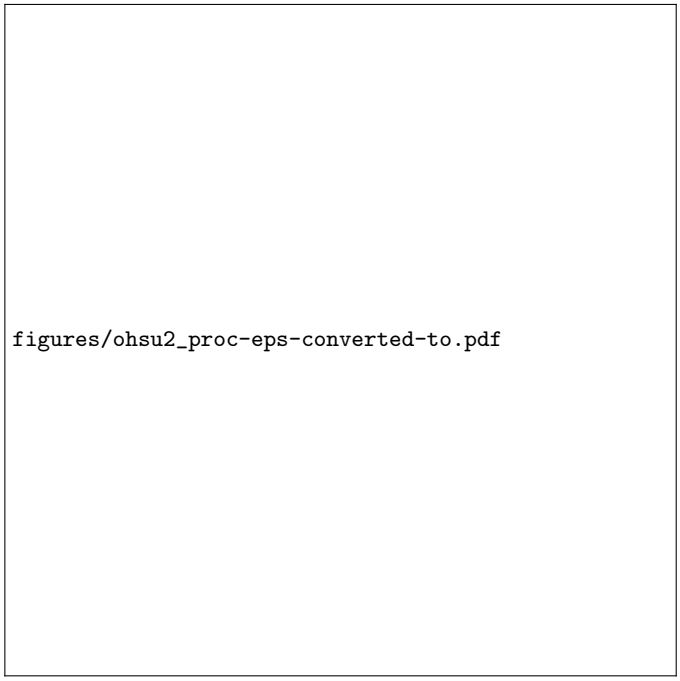

Figure 9 Relative f-measure of the Ohsumed1 test set. The original cosine measure is indicated with a dotted line. The pairwise adaptive dissimilarity measure is shown with a dashed line.

4.4.2 Computational time In this section an overview is given of the computational times for the construction of the dissimilarity matrices. In all figures a clear difference can be seen between the cosine dissimilarity measure and the pairwise adaptive dissimilarity measure. For the smaller test sets Reuters1 and Ohsumed1, related figures

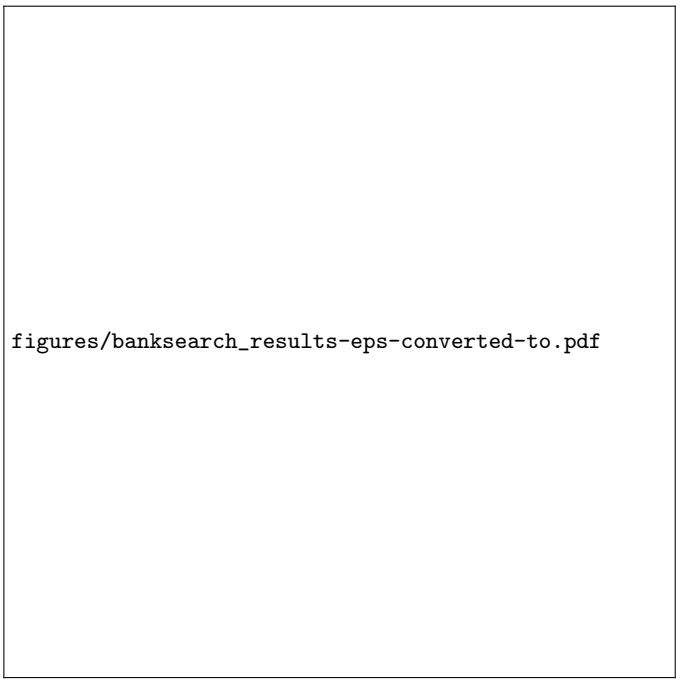

Figure 10 F-measure results of the banksearch test set. The original cosine measure is indicated with a dotted line. The pairwise adaptive dissimilarity measure is shown with a dashed line.

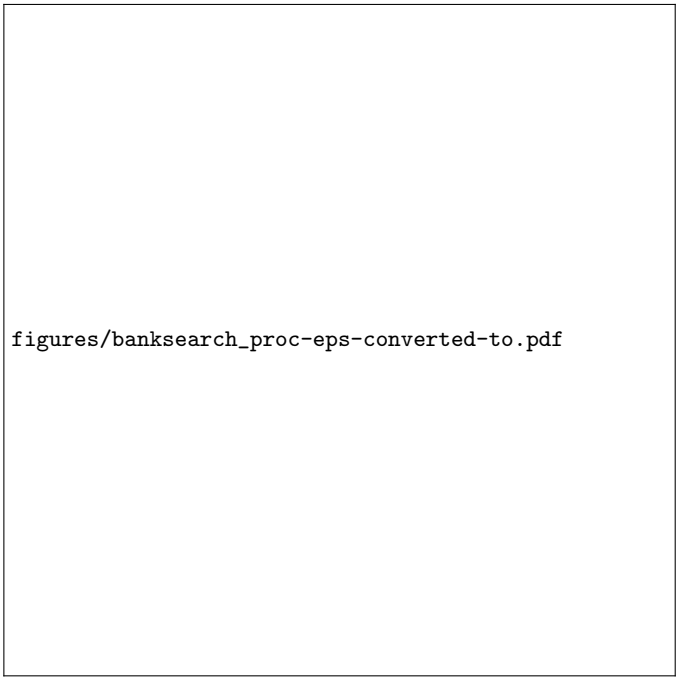

Figure 11 Relative f-measure of the banksearch test set. The original cosine measure is indicated with a dotted line. The pairwise adaptive dissimilarity measure is shown with a dashed line.

?? and **??** show that the pairwise adaptive dissimilarity measure decreases the necessary computational time approximately ten times. For the larger test sets Reuters2 and Ohsumed2, this difference is still a factor of eight. For some test sets of the Banksearch collection **??**, the difference in computational time is even larger.

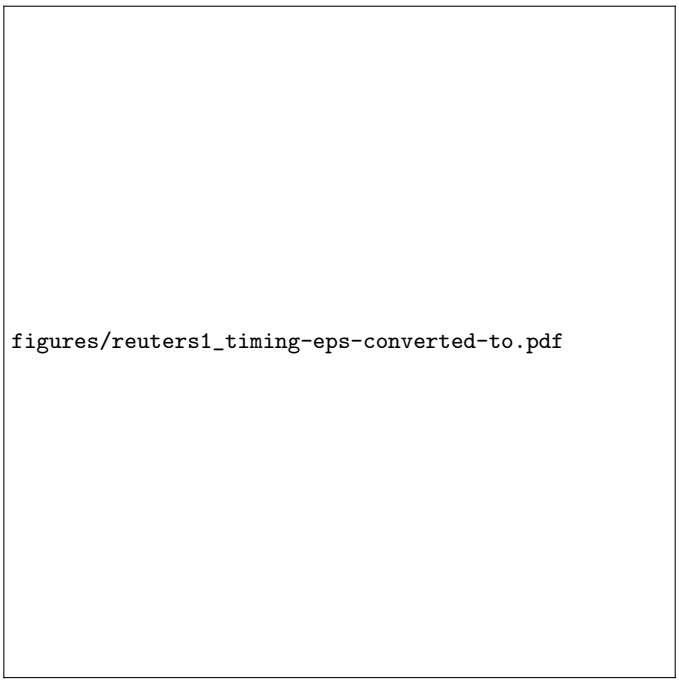

Figure 12 Timing results for constructing the dissimilarity matrices of the Reuters1 test set. The original cosine measure is indicated with a dotted line. The pairwise adaptive dissimilarity measure is shown with a dashed line.

These figures indicate the quadratic nature of the construction of the dissimilarity matrix. To identify the origin of this lower computation time needed for the new dissimilarity measure, the time complexity is further investigated. This research results in an rough upperlimit for the number of terms to be choosen. The computational complexity of the cosine dissimilarity measure between two documents is of a linear nature, formulated as

$$6 * d + 1 \tag{3}$$

where d is the number of dimensions in the vector space model. The computational complexity of the construction of the distance matrix therefor is formulated as

$$(6 * d + 1) * \frac{n * (n-1)}{2} \tag{4}$$

or summarized as

$$O(n^2 \cdot d) \tag{5}$$

The computation complexity of the pairwise adaptive dissimilarity measure is derived as

$$(2 * d * log(d) + 14 * K) \tag{6}$$

where d is the number of dimensions in the vector space model. K is the number of terms taken into account in

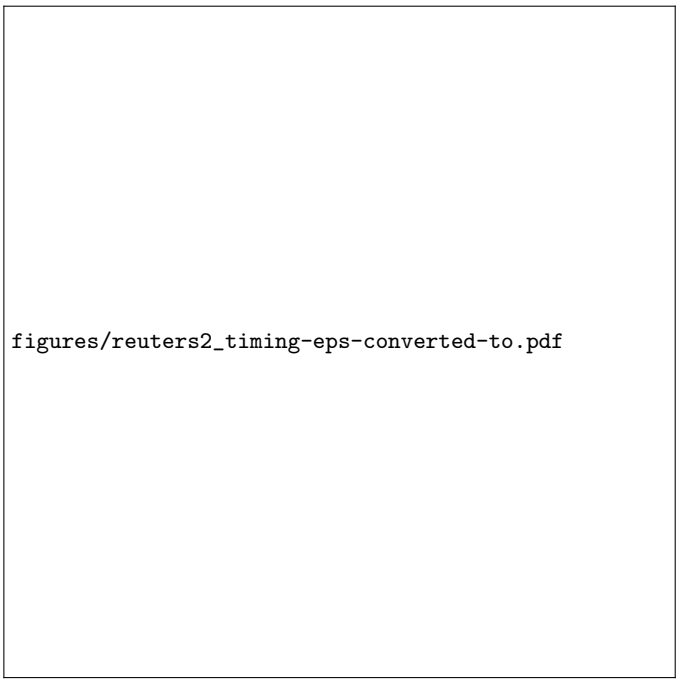

Figure 13 Timing results for constructing the dissimilarity matrices of the Reuters2 test set. The original cosine measure is indicated with a dotted line. The pairwise adaptive dissimilarity measure is shown with a dashed line.

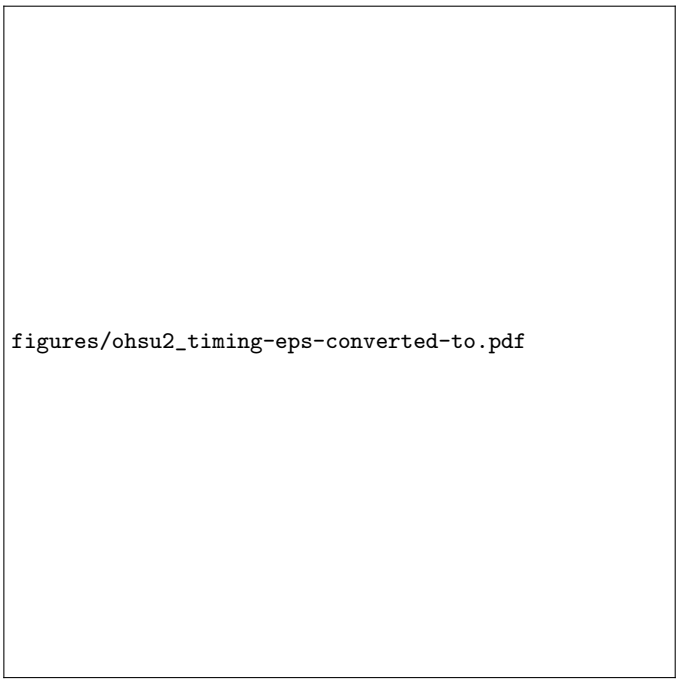

Figure 15 Timing results for constructing the dissimilarity matrices of the Ohsumed2 test set. The original cosine measure is indicated with a dotted line. The pairwise adaptive dissimilarity measure is shown with a dashed line.

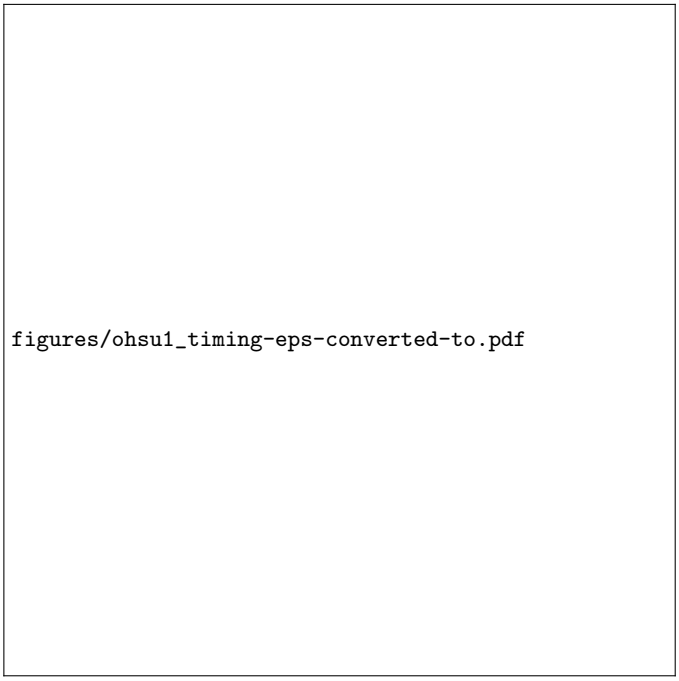

Figure 14 Timing results for constructing the dissimilarity matrices of the Ohsumed1 test set. The original cosine measure is indicated with a dotted line. The pairwise adaptive dissimilarity measure is shown with a dashed line.

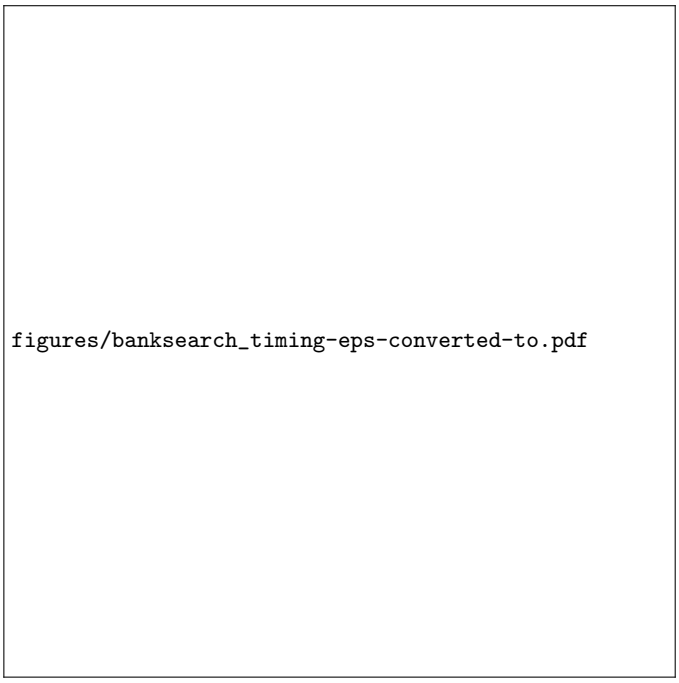

Figure 16 Timing results for constructing the dissimilarity matrices of the banksearch test set. The original cosine measure is indicated with a dotted line. The pairwise adaptive dissimilarity measure is shown with a dashed line.

the dissimilarity calculation. The computational complexity of the construction of the distance matrix therefor is formulated as

$$(2 * d * log(d) + 14 * K) * \frac{n * (n-1)}{2} \qquad (7)$$

Table 4 Computational time in combination with the Singular Value Decomposition

Padist (sec.)	Cosine (sec.)
1,6307	0,0080
10,9493	0,0212
33,7489	0,0454
76,6627	0,0823
146,4718	0,1267
247,1154	0,1764
366,5702	0,2350
541,7428	0,3019
762,1712	0,3820
1050,2000	0,4705

The sorting step is executed one before the actual dissimilarity calculation, therefor the computational complexity can be reduced to

$$n * d * log(d) + \frac{n * (n-1)}{2} + 14 * K \tag{8}$$

or summarized as

$$O(n^2 \cdot K + n \cdot d * log(d)) \tag{9}$$

Comparing equations **??** to **??** indicates that the value for the number of stems K is roughly limited to the value of K resulting in the intersection of these two complexity functions.

4.4.3 Singular Value Decomposition The application of Singular Value Decomposition on the original weighted matrix results in similar F-measure scores for all test sets examined in this research. The application of this feature extraction method however transforms the original sparse matrix to a reduced full matrix. The resulting computational load of applying the pairwise adaptive dissimilarity measure to these matrices is higher compared to load the cosine dissimilarity measure. The computational load for the test set is shown in the table **??**.

This new dissimilarity measure based on a prior feature selection step therefor is not suitable to apply combined with any feature extraction method resulting in a full matrix due to the losing sparseness. [**?**]

5 Summary and conclusions

The presented research defines a new dissimilarity measure applicable in the domain of document clustering. This dissimilarity measure is composed of a feature selection step prior to every dissimilarity measurement. A number of important terms are selected out of every document, and is used to perform dissimilarity measurement. Two approaches of selecting the number of terms based on the document stem density are explained. The dissimilarity technique results in an improved clustering performance compared with the original cosine dissimilarity measure, shown with F-measure in five different test environments. Secondly these results are obtained with a lower computational time needed, therefor creating a considerable speed gain.

A Reuters1

Test set ID	# of topics	# of docs	# of stems
REUTERS2	2	30	2736
REUTERS3	3	45	3921
REUTERS4	4	60	4616
REUTERS5	5	75	5272
REUTERS6	6	90	5990
REUTERS7	7	105	6541
REUTERS8	8	120	7086
REUTERS9	9	135	7648
REUTERS10	10	150	8249
REUTERS11	11	165	8987
REUTERS12	12	180	9488
REUTERS13	13	195	9967
REUTERS14	14	210	10447
REUTERS15	15	225	12255
REUTERS16	16	240	12685
REUTERS17	17	255	13184
REUTERS18	18	270	13633
REUTERS19	19	285	13897

B Reuters2

Test set ID	# of topics	# of docs	# of stems
REUTERS50	4	200	8308
REUTERS100	4	400	11912
REUTERS150	4	600	14801
REUTERS200	4	800	17051
REUTERS250	4	1000	19406
REUTERS300	4	1200	21480
REUTERS350	4	1400	23057
REUTERS400	4	1600	24711
REUTERS450	4	1800	26312
REUTERS500	4	2000	27759
REUTERS550	4	2200	28922
REUTERS600	4	2400	29968
REUTERS650	4	2600	30909
REUTERS700	4	2800	31936
REUTERS750	4	3000	32855
REUTERS800	4	3200	33772
REUTERS850	4	3400	34639
REUTERS900	4	3600	35393
REUTERS950	4	3800	36153
REUTERS1000	4	4000	36890

C Ohsumed1

Test set ID	# of topics	# of docs	# of stems
OHSU2	2	30	645
OHSU3	3	45	852
OHSU4	4	60	1138
OHSU5	5	75	1368
OHSU6	6	90	1524
OHSU7	7	105	1724
OHSU8	8	120	1862
OHSU9	9	135	2026
OHSU10	10	150	2168
OHSU11	11	165	2288
OHSU12	12	180	2409
OHSU13	13	195	2575
OHSU14	14	210	2777
OHSU15	15	225	2906
OHSU16	16	240	3050
OHSU17	17	255	3150
OHSU18	18	270	3244
OHSU19	19	285	3367
OHSU20	20	300	3458
OHSU21	21	315	3602
OHSU22	22	330	3700
OHSU23	23	345	3781
OHSU24	24	360	3884
OHSU25	25	375	3967
OHSU26	26	390	4063
OHSU27	27	405	4147
OHSU28	28	420	4231
OHSU29	29	435	4332
OHSU30	30	450	4439

D Ohsumed2

Test set ID	# of topics	# of docs	# of stems
OHSU50	4	200	4120
OHSU100	4	400	6210
OHSU150	4	600	7810
OHSU200	4	800	9167
OHSU250	4	1000	10286
OHSU300	4	1200	11262
OHSU350	4	1400	12202
OHSU400	4	1600	13034
OHSU450	4	1800	13875
OHSU500	4	2000	14557
OHSU550	4	2200	15218
OHSU600	4	2400	15940
OHSU650	4	2600	16551
OHSU700	4	2800	17168
OHSU750	4	3000	17726
OHSU800	4	3200	18255
OHSU850	4	3400	18723
OHSU900	4	3600	19231
OHSU950	4	3800	19664
OHSU1000	4	4000	20104

E Banksearch

Test set ID	# of topics	# of docs	# of stems
Banksearch1	10	1257	16108
Banksearch2	5	490	21014
Banksearch3	4	305	36326
Banksearch4	4	331	17346
Banksearch5	5	268	23450
Banksearch6	5	245	15317
Banksearch7	2	151	12953
Banksearch8	10	430	18942
Banksearch9	3	226	17531
Banksearch10	4	197	18660
Banksearch11	4	218	9739
Banksearch12	4	256	8643
Banksearch13	5	386	14743
Banksearch14	2	165	9373
Banksearch15	3	346	9029
Banksearch16	5	592	13395
Banksearch17	10	927	17136
Banksearch18	4	309	9086
Banksearch19	5	437	7615
Banksearch20	3	281	7388
Banksearch21	9	2124	10679