Amélioration de la co-similarité pour la classification de documents

Clément Grimal[1], Gilles Bisson[1]

Laboratoire d'Informatique de Grenoble, UMR 5217
Clement.Grimal@imag.fr, Gilles.Bisson@imag.fr

Abstract : La classification conjointe d'objets et de leurs descripteurs – par exemple de documents avec les mots les composant – encore appelée *co-classification*, a été largement étudiée ces dernières années, car elle permet d'extraire des classes plus pertinentes, qu'elle soit explicite ou latente. Dans de précédents travaux (Bisson & Hussain, 2008), nous avons proposé une méthode de calcul simultané des matrices de similarité entre objets et entre descripteurs, chacune étant construite à partir de l'autre. Nous proposons ici une généralisation de cette approche en introduisant une pseudo-norme et un algorithme de seuillage. Nos expérimentations mettent en évidence une amélioration significative de la classification.

Mots-clés : co-clustering, similarity measure, text mining

1. Introduction

Clustering task is used to organize data coming from databases. Classically, these data are described as a set of instances characterized by a set of features. In some cases, these features are homogeneous enough to allow us to cluster them, in the same way as we do for the instances. For example, when using the *Vector Space Model* introduced by Salton (1971), text corpora are represented by a matrix whose rows represent document vectors and whose columns represent the word vectors. Thus, the similarity between two documents depends on the similarity between the words they contain and vice-versa. In the classical clustering methods, such dependencies are not exploited. The purpose of co-clustering is to take into account this duality between rows and columns to identify the relevant clusters. Co-clustering has been largely studied in recent years both in *Document clustering* (Dhillon *et al.*, 2003; Long *et al.*, 2005;

Rege *et al.*, 2008; Liu *et al.*, 2004) and *Bioinformatics* (Madeira & Oliveira, 2004; Speer *et al.*, 2004; Cheng & Church, 2000).

In text analysis, the advantage of co-clustering is related to the well-known problem that document and words vectors tend to be highly sparse and suffer from the curse of dimensionality (Slonim & Tishby, 2001). Thus, traditional metrics such as Euclidean distance or Cosine similarity do not always make much sense (Beyer *et al.*, 1999). Several methods have been proposed to overcome these limitations by exploiting the dual relationship between documents and words to extract semantic knowledge from the data. Consequently, the concept of *higher-order* co-occurrences has been investigated by Livezay & Burgess (1998); Lemaire & Denhière (2008), among others, as a measure of semantic relationship between words; one of the best known approach to acquire such knowledge being the Latent Semantic Analysis (Deerwester *et al.*, 1990). The underlying analogy is that humans do not necessarily use the same vocabulary when writing about the same topic. For example, let us consider a corpus in which a subset of documents contains a significant number of co-occurrences between the words *sea* and *waves* and another subset in which the words *ocean* and *waves* co-occur. A human could infer that the worlds *ocean* and *sea* are conceptually related even if they do not directly co-occur in any document. Such a relationship between *waves* and *ocean* (or *sea* and *waves*) is termed as a *first-order* co-occurrence and the conceptual association between *sea* and *ocean* is called a *second-order* relationship. This concept can be generalized to higher-order (3$^{\text{rd}}$, 4$^{\text{th}}$, 5$^{\text{th}}$, etc) co-occurrences.

In this context, we recently introduced an algorithm, called χ-Sim (Bisson & Hussain, 2008), exploiting the duality between words and documents in a corpus as well as their respective higher-order co-occurrences. While most authors have focused to directly co-cluster the data, in χ-Sim, we build two similarity matrices, one for the rows and one for the columns, each being built iteratively on the basis of the other. Hence, when the two similarity matrices are built, each of them contains all the information needed to do a 'separate' co-clustering of the data (documents and words) by using any classical clustering algorithm.

In this paper, we further analyze the behavior of χ-Sim and we propose some ideas leading to dramatically improve the quality of the co-similarity measures. First, we introduce a new normalization schema for this measure that is more consistent with the framework of the algorithm and that offers new perspectives of research. Second, we propose an efficient way to deal with noise in the data and thus to improve the accuracy of the clustering.

2. The χ-Sim Similarity Measure

Throughout this paper, we will use the classical notations: matrices (in capital letters) and vectors (in small letters) are in bold and variables are in italic.
Data matrix: let $\mathbf{M}$ be the data matrix representing a corpus having r rows (documents) and c columns (words); m_{ij} corresponds to the 'intensity' of the link between the i^{th} row and the j^{th} column (for a document-word matrix, it can be the frequency of the j^{th} word in the i^{th} document); $\mathbf{m}_{i:} = [m_{i1} \cdots m_{ic}]$ is the row vector representing the document i and $\mathbf{m}_{:j} = [m_{1j} \cdots m_{rj}]$ is the column vector corresponding to word j. We will refer to a document as d_i when talking about documents casually and refer to it as $\mathbf{m}_{i:}$ when specifying its (row) vector in the matrix $\mathbf{M}$. Similarly, we will casually refer to a word as w_j and use the notation $\mathbf{m}_{:j}$ when emphasizing the vector.
Similarity matrices: $\mathbf{SR}$ and $\mathbf{SC}$ represent the square and symmetrical row similarity and column similarity matrices of size $r \times r$ and $c \times c$ respectively, with $\forall i, j = 1..r, sr_{ij} \in [0,1]$ and $\forall i, j = 1..c, sc_{ij} \in [0,1]$.
Similarity function: generic function $\mathrm{F_s}(\cdot,\cdot)$ takes two elements m_{il} and m_{jn} of $\mathbf{M}$ and returns a measure of the similarity $\mathrm{F_s}(m_{il}, m_{jn})$ between them.

2.1. Similarity measures

χ-Sim is a co-similarity based approach which builds on the idea of simultaneously generating the similarity matrices $\mathbf{SR}$ (documents) and $\mathbf{SC}$ (words), each of them built on the basis of the other. Similar ideas have also been used for supervised leaning in (Liu *et al.*, 2004) or for image retrieval in (Wang *et al.*, 2004). First, we present how to compute the matrix $\mathbf{SR}$. Usually, the similarity (or distance) measure between two documents $\mathbf{m}_{i:}$ and $\mathbf{m}_{j:}$ is defined as a function – denoted here as $\mathrm{Sim}(\mathbf{m}_{i:}, \mathbf{m}_{j:})$ – that is more or less the sum of the similarities between words occurring in both $\mathbf{m}_{i:}$ and $\mathbf{m}_{j:}$:

$$\mathrm{Sim}(\mathbf{m}_{i:}, \mathbf{m}_{j:}) = \mathrm{F_s}(m_{i1}, m_{j1}) + \cdots + \mathrm{F_s}(m_{ic}, m_{jc}) \tag{1}$$

Now let's suppose we already know a matrix $\mathbf{SC}$ whose entries provide a measure of similarity between the columns (words) of the corpus. In parallel, let's introduce, by analogy to the norm L_k (Minkowski distance), the notion of a *pseudo-norm* k. Then (1) can be re-written as follows without changing its meaning if $\mathbf{sc}_{ll} = 1$ and if $k = 1$:

$$\mathrm{Sim}(\mathbf{m}_{i:}, \mathbf{m}_{j:}) = \sqrt[k]{\sum_{l=1}^{c} \left(\mathrm{F_s}(m_{il}, m_{jl})\right)^k \times sc_{ll}} \tag{2}$$

Now the idea is to generalize (2) in order to take into account all the possible pairs of features (words) occurring in documents $\mathbf{m}_{i:}$ and $\mathbf{m}_{j:}$. In this way, we "capture" not only the similarity between their common words but also the similarity coming from words *that are not directly shared* by the two documents. Of course, for each pair of words not directly shared by the documents, we weight their contribution to the document similarity sr_{ij} by their own similarity sc_{ln}. Thus, the overall similarity between documents $\mathbf{m}_{i:}$ and $\mathbf{m}_{j:}$ is defined in (3) in which the terms for $l = n$ are those occurring in (2):

$$\mathrm{Sim}^k(\mathbf{m}_{i:}, \mathbf{m}_{j:}) = \sqrt[k]{\sum_{l=1}^{c} \sum_{n=1}^{c} \left(\mathrm{F_s}\left(m_{il}, m_{jn}\right)\right)^k \times sc_{ln}} \tag{3}$$

Assuming that $\mathrm{F_s}(m_{il}, m_{jn})$ is defined as a product (see (Bisson & Hussain, 2008) for further details) of the elements m_{il} and m_{jn}, i.e. $\mathrm{F_s}(m_{il}, m_{jn}) = m_{il} \times m_{jn}$ (as with the cosine similarity), we can rewrite (3) as:

$$\mathrm{Sim}^k(\mathbf{m}_{i:}, \mathbf{m}_{j:}) = \sqrt[k]{(\mathbf{m}_{i:})^k \times \mathbf{SC} \times \left(\mathbf{m}_{j:}^{\mathrm{T}}\right)^k} \tag{4}$$

where $(\mathbf{m}_{i:})^k = \left[(m_{ij})^k \cdots (m_{ic})^k\right]$ and $\mathbf{m}_{j:}^{\mathrm{T}}$ denotes the transpose of the vector $\mathbf{m}_{j:}$. Finally, let's introduce the term $\mathcal{N}(\mathbf{m}_{i:}, \mathbf{m}_{j:})$ that is a normalization function allowing to map the similarity to $[0, 1]$. We obtain the following equation in which sr_{ij} denotes an element of the $\mathbf{SR}$ matrix:

$$sr_{ij} = \frac{\sqrt[k]{(\mathbf{m}_{i:})^k \times \mathbf{SC} \times \left(\mathbf{m}_{j:}^{\mathrm{T}}\right)^k}}{\mathcal{N}(\mathbf{m}_{i:}, \mathbf{m}_{j:})} \tag{5}$$

Equation (5) is a classic generalization of several well-known similarity measures. For example, with $k = 1$, the Jacard index can be obtained by setting $\mathbf{SC}$ to $\mathbf{I}$ and $\mathcal{N}(\mathbf{m}_{i:}, \mathbf{m}_{j:})$ to $\|\mathbf{m}_{i:}\|_1 + \|\mathbf{m}_{j:}\|_1 - \mathbf{m}_{i:}\mathbf{m}_{j:}^{\mathrm{T}}$, while the Dice coefficient can be obtained by setting $\mathbf{SC}$ to $2\mathbf{I}$ and $\mathcal{N}(\mathbf{m}_{i:}, \mathbf{m}_{j:})$ to $\|\mathbf{m}_{i:}\|_1 + \|\mathbf{m}_{j:}\|_1$. Furthermore, if $\mathbf{SC}$ is set to a *positive semi-definite* matrix $\mathbf{A}$, one can define the following inner product $< \mathbf{m}_{i:}, \mathbf{m}_{j:} >_{\mathbf{A}} = \mathbf{m}_{i:} \times A \times \mathbf{m}_{j:}^{\mathrm{T}}$, along with the associated norm $\|\mathbf{m}_{i:}\|_{\mathbf{A}} = < \mathbf{m}_{i:}, \mathbf{m}_{i:} >_{\mathbf{A}}$. Then by setting $\mathcal{N}(\mathbf{m}_{i:}, \mathbf{m}_{j:})$ to $\sqrt{\|\mathbf{m}_{i:}\|_{\mathbf{A}}} \times \sqrt{\|\mathbf{m}_{j:}\|_{\mathbf{A}}}$, we obtain the Generalized Cosine similarity (Qamar & Gaussier, 2009), as it corresponds to the Cosine measure in the underlying inner product space. Of course, by binding $\mathbf{A}$ to $\mathbf{I}$, this similarity becomes the standard Cosine measure between document $\mathbf{m}_{i:}$ and $\mathbf{m}_{j:}$.

2.2. The χ-Sim Co-Similarity Measure

Of course, χ-Sim co-similarity, as defined in (Bisson & Hussain, 2008) can be also reformulated with (5). We set k to 1, and since the maximum value defined by the function sc_{ij} is 1, it follows from (3) that, the upper bound of $\mathrm{Sim}(\mathbf{m}_{i:}, \mathbf{m}_{j:})$ for $1 \leqslant i, j \leqslant r$, is given by the product of the sum of elements of $\mathbf{m}_{i:}$ and $\mathbf{m}_{j:}$ denoted by $|\mathbf{m}_{i:}| \times |\mathbf{m}_{j:}|$ (product of L$_1$-norms). This normalization seems well-suited for textual datasets since it allows us to take into consideration pairs of documents (or words) vectors of uneven length, which is common in text corpora. Therefore, we can rewrite (5) as:

$$\forall i, j \in 1..r, sr_{ij} = \frac{\mathbf{m}_{i:} \times \mathbf{SC} \times \mathbf{m}_{j:}^{\mathrm{T}}}{|\mathbf{m}_{i:}| \times |\mathbf{m}_{j:}|} \tag{6a}$$

Symmetrically, the elements sc_{ij} of the $\mathbf{SC}$ matrix are defined as:

$$\forall i, j \in 1..c, sc_{ij} = \frac{\mathbf{m}_{:i}^{\mathrm{T}} \times \mathbf{SR} \times \mathbf{m}_{:j}}{|\mathbf{m}_{:i}| \times |\mathbf{m}_{:j}|} \tag{6b}$$

Equations (6a) and (6b) define a systems of linear equations, whose solutions correspond to the (co)-similarities between two documents and two words. Thus, the algorithm of χ-Sim is based on an iterative approach – i.e. we compute alternatively the values sc_{ij} and sr_{ij}. However, before detailing this algorithm for a more generic case in section 3.3., we are going to explain the meaning, considering the associated bipartite graph, of these iterations.

2.3. Graph Theoretical Interpretation

The graphical interpretation of the method helps to understand the working of the algorithm and provides some intuition on how to improve it. Let's consider the bipartite graph representation of a sample data matrix in Fig. 1. Documents and words are represented by square and circle nodes respectively and an edge (of any kind) between a document d_i and a word w_j corresponds to a non-zero entry m_{ij} in the document-word matrix. There is only one order-1 path between documents d_1 and d_2 given by $\mathrm{d}_1 \xrightarrow{m_{12}} \mathrm{w}_2 \xrightarrow{m_{22}} \mathrm{d}_2$. If we consider that the $\mathbf{SC}$ matrix is initialized with the identity matrix $\mathbf{I}$, at the first iteration, $\mathrm{Sim}(\mathbf{m}_{1:}, \mathbf{m}_{2:})$ corresponds to the inner product between $\mathbf{m}_{1:}$ and $\mathbf{m}_{2:}$ as given by (6a) and equals $m_{12} \times m_{22}$. Omitting the normalization for the sake of clarity, the matrix $\mathbf{SR}^{(1)} = \mathbf{M} \times \mathbf{M}^{\mathrm{T}}$ thus represents all order-1 paths between all the possible pairs of documents d_i and d_j. Similarly, each element of $\mathbf{SC}^{(1)} = \mathbf{M}^{\mathrm{T}} \times \mathbf{M}$ represents all order-1 paths between all the possible pairs of words w_i and w_j. Now, documents d_1 and d_4 do not have an order-1 path but are linked together through d_2 (bold paths in Fig. 1) and d_3 (dotted paths in Fig. 1). Such paths with one intermediate vertice are called order-2 paths, and will appear during the second iteration. The similarity value contributed via the document d_2 can be explicitly represented as $\mathrm{d}_1 \xrightarrow{m_{12}} \mathrm{w}_2 \xrightarrow{m_{22}} \mathrm{d}_2 \xrightarrow{m_{24}} \mathrm{w}_4 \xrightarrow{m_{44}} \mathrm{d}_4$. The sub-sequence $\mathrm{w}_2 \xrightarrow{m_{22}} \mathrm{d}_2 \xrightarrow{m_{24}} \mathrm{w}_4$ represents an order-1 path between words w_2 and w_4 which value is $sc_{24}^{(1)}$. The contribution of d_2 in the similarity of $sr_{14}^{(1)}$ can thus be re-written as $m_{12} \times sc_{24}^{(1)} \times m_{44}$. This is a partial similarity measure since d_2 is not the only document that provides a link between d_1 and d_4. The similarity via d_3 is equal to $m_{13} \times sc_{35}^{(1)} \times m_{55}$. To find the overall similarity measure between documents d_1 and d_4, we need to add these partial similarity values given by $m_{12} \times sc_{24}^{(1)} \times m_{44} + m_{13} \times sc_{35}^{(1)} \times m_{55}$. Hence, the similarity matrix $\mathbf{SR}^{(2)}$ at the second iteration corresponds to paths of order-2 between documents. It can be shown similarly that, the matrices $\mathbf{SR}^{(t)}$ and $\mathbf{SC}^{(t)}$ represent order-t paths between documents and between words respectively.

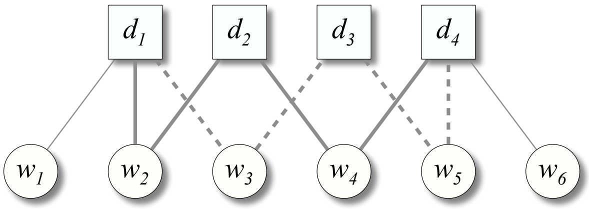

Figure 1: A bi-partite graph view of a sample document-word matrix.

Consequently, at iteration t, when we compute the value of equations (6a) and (6b), one or more new links may be found between previously disjoint objects (documents or words) corresponding to paths with length of order-t, and existing similarity measures may be strengthened. It has been shown that “in the long run”, the ending point of a random walk does not depend on its starting point (Seneta, 2006) and hence it is possible to find a path (and hence similarity) between any pair of nodes in a connected graph (Zelikovitz & Hirsh, 2001) by iterating a sufficiently large number of times. However, co-occurrences beyond the 3rd and 4th order have little semantic relevance (Lemaire & Denhière, 2008). Therefore, the number of iterations is usually limited to 4 or less.

3. Discussion and Improvements of χ-Sim

In this section, first, we discuss a new normalization schema for χ-Sim in order to (partially) satisfy the maximality property of a similarity measure ($\text{Sim}(a, b) = 1$), then we propose a pruning method allowing to remove the 'noisy' similarity values created during the iterations.

3.1. Normalization

In this paper, we investigate extensions of the Generalized Cosine measure, by relaxing the *positive semi-definiteness* of the matrix, and by adding a pseudo-norm parameter k. Henceforth, using the (4) we define the elements of the matrices $\mathbf{SR}$ and $\mathbf{SC}$:

$$\forall i, j \in 1..r,\ sr_{ij} = \frac{\text{Sim}^k(\mathbf{m}_{i:}, \mathbf{m}_{j:})}{\sqrt{\text{Sim}^k(\mathbf{m}_{i:}, \mathbf{m}_{i:})} \times \sqrt{\text{Sim}^k(\mathbf{m}_{j:}, \mathbf{m}_{j:})}} \tag{7a}$$

$$\forall i, j \in 1..c,\ sc_{ij} = \frac{\text{Sim}^k(\mathbf{m}_{:i}, \mathbf{m}_{:j})}{\sqrt{\text{Sim}^k(\mathbf{m}_{:i}, \mathbf{m}_{:i})} \times \sqrt{\text{Sim}^k(\mathbf{m}_{:j}, \mathbf{m}_{:j})}} \tag{7b}$$

However, this normalization is what we will call a *pseudo-normalization* since it guaranties that $sr_{ii} = 1$, but it does not satisfy that $\forall i, j \in 1..r, sr_{ij} \in [0, 1]$. Consider for example a corpus having, among other documents, the documents d_1 containing the word *orange* (w_1) and d_2 containing the words *red* (w_2) and *banana* (w_3), along with $\mathbf{SC}$ – the similarity matrix of all the words of the corpus – indicating that the similarity between *orange* and *red* is 1, the similarity between *orange* and *banana* is 1 and the similarity between *red* and *banana* is 0. Thus, $\text{Sim}^1(\mathrm{d}_1, \mathrm{d}_1) = 1$, $\text{Sim}^1(\mathrm{d}_2, \mathrm{d}_2) = 2$ and $\text{Sim}^1(\mathrm{d}_1, \mathrm{d}_2) = 2$. Consequently, $sr_{12} = \frac{2}{\sqrt{1 \times 2}} > 1$. One can notice that this problem arises from the polysemic nature of the word *orange*. Indeed, the similarity between these two documents is overemphasized because of the double analogy between *orange* (the color) and *red*, and between *orange* (the fruit) and *banana*. It is possible to correct this problem by setting $k = +\infty$ since the pseudo-norm-k becomes $\max_{1 \leqslant k, l \leqslant c} \{m_{ik} \times sc_{kl} \times m_{il}\}$ and thus, $\text{Sim}^\infty(\mathrm{d}_1, \mathrm{d}_1) = \text{Sim}^\infty(\mathrm{d}_2, \mathrm{d}_2) = \text{Sim}^\infty(\mathrm{d}_1, \mathrm{d}_2) = 1$, implying $sr_{12} = 1$. Of course, $k = +\infty$ is not necessarily a good setting for real tasks and experimentally we observed that the values of sr_{ij} and sc_{ij} remain generally smaller or equal to 1.

In this framework it is nevertheless very interesting to investigate the different results one can obtain from varying k, including values lower than 1, as suggested by Aggarwal *et al.* (2001) for the norm L_k, to deal with high dimensional spaces. The resulting χ-Sim algorithm will be denoted by χ-Sim^k. However, the situation is different from the norm L_k in the sense that our method does not define a proper Normed Vector Space. To understand the problem it is worth looking closely to the simple case $k = 1$ where $\mathrm{Sim}^1(\mathbf{m}_{i:}, \mathbf{m}_{j:}) = \mathbf{m}_{i:} \times \mathbf{SC} \times \mathbf{m}_{i:}^{\mathrm{T}}$ is the general form of an inner product, with the condition that $\mathbf{SC}$ is symmetric *positive semi-definite (PSD)*. Unfortunately, in our case due to the normalization steps, $\mathbf{SC}$ is not necessarily PSD, as the condition $\forall i, j \in 1..c, |sc_{ij}| \leqslant \sqrt{sc_{ii} \times sc_{jj}} = 1$ is not verified (cf. previous example). Thus, our similarity measure is just a bilinear form in a degenerated inner product space (as the conjugate and linearity axioms are trivially satisfied) in which it corresponds to the cosine.

A straightforward solution would be to project $\mathbf{SC}$ (and $\mathbf{SR}$) after each iteration onto the set of PSD matrices (Qamar & Gaussier, 2009). By constraining the similarity matrices to be PSD, we would ensure that the new space remains a proper inner product space. However, we experimentally verified that such an additional step did not improve the results though, as when testing on real datasets, the similarity matrices are very close to the set of PSD matrices. In addition, the projection step is very time consuming, for these reasons, we won't consider it in the remaining of this paper.

3.2. Dealing with 'noise' in SC and SR matrices

As explained in section 2.3., the elements of the $\mathbf{SR}$ matrix after the first iteration are the weighted order-1 paths in the graph: the diagonal elements sr_{ii} correspond to the paths from each document to itself, while the non-diagonal terms sr_{ij} count the number of order-1 paths between a document i and a neighbour j, which is based on the number of words they have in common. $\mathbf{SR}^{(1)}$ is thus the adjacency matrix of the document graph. Iteration t amounts thus to count the number of order-t between nodes. However, in a corpus, we can observe there are many words with few occurrences, that are not really relevant with the topic of the document, or to be more precise, that are not specific to any *families of documents semantically related*. These words are similar to a 'noise' in the dataset. Thus, during the iterations, these noisy words allow the algorithm to create some new paths between the different families of documents; of course these paths have a very small similarity value but they are numerous and we make the assumption that they blur the similarity values between the classes of documents (same for the words). Based on this observation, we thus introduce in the χ-Sim algorithm a parameter, termed *pruning threshold* and denoted by p, allowing us to set to zero the lowest $p\,\%$ of the similarity values in the matrices $\mathbf{SR}$ and $\mathbf{SC}$ at each iteration.

In the following, we will refer to this algorithm as χ-Sim$_p$ when using the previous normalization factor described in (6a) and (6b), and χ-Sim$_p^k$ when using the new pseudo-normalization factor described in (7a) and (7b).

3.3. A Generic χ-Sim Algorithm for χ-Sim$_p^k$

Equations (7a) and (7b) allows us to compute the similarities between two rows and two columns. The extension over all pair of rows and all pair of columns can be generalized under the form of a simple matrix multiplication. We need to introduce a new notation here, $\mathbf{M}^{\circ k} = \left((m_{ij})^k\right)_{i,j}$ which is the element-wise exponentiation of $\mathbf{M}$ to the power of k. The algorithm follows:

1. We initialize the similarity matrices $\mathbf{SR}$ (documents) and $\mathbf{SC}$ (words) with the identity matrix $\mathbf{I}$, since, at the first iteration, only the similarity between a row (resp. column) and itself equals 1 and zero for all other rows (resp. columns). We denote these matrices as $\mathbf{SR}^{(0)}$ and $\mathbf{SC}^{(0)}$.

2. At iteration t, we calculate the new similarity matrix between documents $\mathbf{SR}^{(t)}$ by using the similarity matrix between words $\mathbf{SC}^{(t-1)}$:

$$\mathbf{SR}^{(t)} = \mathbf{M}^{\circ k} \times \mathbf{SC}^{(t-1)} \times (\mathbf{M}^{\circ k})^{\mathrm{T}} \text{ and } sr_{ij}^{(t)} \leftarrow \frac{\sqrt[k]{sr_{ij}^{(t)}}}{\sqrt[2k]{sr_{ii}^{(t)} \times sr_{jj}^{(t)}}} \quad (8)$$

   We do the same thing for the columns similarity matrix $\mathbf{SC}^{(t)}$:

$$\mathbf{SC}^{(t)} = (\mathbf{M}^{\circ k})^{\mathrm{T}} \times \mathbf{SR}^{(t-1)} \times \mathbf{M}^{\circ k} \text{ and } sc_{ij}^{(t)} \leftarrow \frac{\sqrt[k]{sc_{ij}^{(t)}}}{\sqrt[2k]{sc_{ii}^{(t)} \times sc_{jj}^{(t)}}} \quad (9)$$

3. We set to 0 the $p\,\%$ of *the lowest similarity values* in the similarity matrices $\mathbf{SR}$ and $\mathbf{SC}$.

4. Steps 2 and 3 are repeated t times (typically as we saw in section 2, 4 iterations are enough) to iteratively update $\mathbf{SR}^{(t)}$ and $\mathbf{SC}^{(t)}$.

It is worth noting here that even though χ-Sim$_p^k$ computes the similarity between each pair of documents using all pairs of words, the complexity of the algorithm remains comparable to classical similarity measures like Cosine. Given that – for a generalized matrix of size n by n – the complexity of matrix multiplication is in $\mathcal{O}(n^3)$ and the complexity to compute $\mathbf{M}^{\circ n}$ is in $\mathcal{O}(n^2)$, the overall complexity of χ-Sim is given by $\mathcal{O}(tn^3)$.

4. Experiments

Here, to evaluate our system, we cluster the documents coming from the well-known 20-Newsgroup dataset (NG20) by using the documents similarity matrices $\mathbf{SR}$ generated by χ-Sim. We choose this dataset since it has been widely used as a benchmark for document classification and co-clustering (Dhillon *et al.*, 2003; Long *et al.*, 2005; Zhang *et al.*, 2007; Long *et al.*, 2006), thus allowing us to compare our results with those reported in the literature.

4.1. Preprocessing and Methodology

Test dataset. We replicate the experimental procedures used by Dhillon *et al.* (2003); Long *et al.* (2005, 2006): 10 different samples of each of the 6 subsets described in Table 1 are generated, we ignored the subject lines, we removed stop words and we selected the top 2,000 words based on *supervised mutual information* (Yang & Pedersen, 1997). We will discuss further this last preprocessing step in section 5. With these six benchmarks, we compared our co-similarity measures based on χ-Sim with four similarity measures: **Cosine**, **LSA** (Deerwester *et al.*, 1990), **SNOS** (Liu *et al.*, 2004) and **CTK** (Yen *et al.*, 2009); as well as three co-clustering methods: **ITCC** (Dhillon *et al.*, 2003), **BVD** (Long *et al.*, 2005) and **RSN** (Long *et al.*, 2006).

Table 1: Description of the subsets of the NG20 dataset used. We provide the number of clusters, and the number of documents for every subset.

	Newsgroups included	#clust.	#docs.
M2	talk.politics.mideast, talk.politics.misc	2	500
M5	comp.graphics, rec.motorcycles, rec.sport.baseball, sci.space, talk.politics.mideast	5	500
M10	alt.atheism, comp.sys.mac.hardware, misc.forsale, rec.autos, rec.sport.hockey, sci.crypt, sci.electronics, sci.med, sci.space, talk.politics.gun	10	500
NG1	rec.sports.baseball, rec.sports.hockey	2	400
NG2	comp.os.ms-windows.misc, comp.windows.x, rec.motorcycles, sci.crypt, sci.space	5	1000
NG3	comp.os.ms-windows.misc, comp.windows.x, misc.forsale, rec.motorcycles, sci.crypt, sci.space, talk.politics.mideast, talk.religion.misc	8	1600

Creation of the clusters. For the 'similarity based' algorithms: χ-Sim, Cosine, LSA, SNOS and CTK, the clusters are generated by an *Agglomerative Hierarchical Clustering* (AHC) method with Ward's linkage on $\mathbf{SR}$. Then we cut the clustering tree at the level corresponding to the number of document clusters we are waiting for (two for subset M2, etc).

Implementations. χ-Sim algorithms, as well as Cosine, SNOS and AHC have been implemented in Python, and LSA have been implemented in MatLab. For CTK and ITCC, we used the implementations provided by the authors.For BVD and RSN, as we don't have a running implementation, we directly quote the best values from (Long *et al.*, 2005) and (Long *et al.*, 2006) respectively.

4.2. Experimental Measures

We used the classical *micro-averaged precision* (Pr) (Dhillon *et al.*, 2003) to compare the accuracy of the document classification; the Normalized Mutual Information (NMI) (Banerjee & Ghosh, 2002) is also used to compare χ-Sim with RSN. For SNOS, we perform four iterations and set the λ parameter to the value proposed by Liu *et al.* (2004). For LSA, we tested the algorithm iteratively keeping the h highest singular values from $h = 10..200$ by steps of 10. We use the value of h providing, on average, the highest micro-averaged precision. For CTK, we used the sigmoid transformation with the parameter $a = 7$. For ITCC, we ran three times the algorithm using the different numbers of word clusters, as suggested by Dhillon *et al.* (2003), for each dataset. For χ-Sim$_p$, we performed the pruning step as described in section 3.3. varying the value of $p = 0$ to 0.9 by steps of 0.1. For each subset, we report the best micro-averaged precision obtained with p.

The experimental results are summarized in Table 2. In all the versions, χ-Sim performs better than all the other tested algorithms. Moreover, the new normalization schema proposed in Section 3. clearly improved the results of our algorithm over the previous normalization based on the length of the documents. The SNOS algorithm performs poorly in spite of the fact that it is very closed to χ-Sim, probably because it uses a different normalization. It is interesting to notice that the gain obtained with the pruning when using the previous version of χ-Sim on the M10 and NG3 (the two hardest problems) is reduced to almost negligible levels with the new algorithm. Finally, the impact of the parameter k is small for all the subsets but M10 and NG3. In these more complex datasets, we observe that setting k to a value lower than

1 slightly improves the clustering but not in a significative way. This result seems to show that the results provided by (Aggarwal *et al.*, 2001), suggesting to use a value of k lower than 1 with the norm L_k when dealing with high dimensional space, could be relevant in our framework (see next section).

Table 2: Micro-averaged precision (and NMI for χ-Sim based algorithms and RSN) and standard deviation for the various subsets of NG20.

		M2	M5	M10	NG1	NG2	NG3
Cosine	Pr	0.60 ± 0.00	0.63 ± 0.07	0.49 ± 0.06	0.90 ± 0.11	0.60 ± 0.10	0.59 ± 0.04
LSA	Pr	0.92 ± 0.02	0.87 ± 0.06	0.59 ± 0.07	0.96 ± 0.01	0.82 ± 0.03	0.74 ± 0.03
ITCC	Pr	0.79 ± 0.06	0.49 ± 0.10	0.29 ± 0.02	0.69 ± 0.09	0.63 ± 0.06	0.59 ± 0.05
BVD	Pr	best: 0.95	best: 0.93	best: 0.67	-	-	-
RSN	NMI	-	-	-	0.64 ± 0.16	0.75 ± 0.07	0.70 ± 0.04
SNOS	Pr	0.55 ± 0.02	0.25 ± 0.02	0.24 ± 0.06	0.51 ± 0.01	0.24 ± 0.02	0.22 ± 0.05
CTK	Pr	0.94 ± 0.01	0.95 ± 0.01	0.71 ± 0.03	0.96 ± 0.01	0.90 ± 0.01	0.87 ± 0.02
χ-Sim	Pr NMI	0.91 ± 0.09	0.96 ± 0.00	0.69 ± 0.05	0.96 ± 0.01 0.76 ± 0.06	0.92 ± 0.01 0.79 ± 0.02	0.79 ± 0.06 0.72 ± 0.03
χ-Sim$_p$	Pr NMI	0.94 ± 0.01	0.96 ± 0.00	0.73 ± 0.03	0.97 ± 0.01 0.78 ± 0.05	0.92 ± 0.01 0.79 ± 0.02	0.84 ± 0.05 0.73 ± 0.02
χ-Sim^1	Pr NMI	$\mathbf{0.95} \pm 0.00$	0.96 ± 0.02	0.78 ± 0.03	0.97 ± 0.02 0.85 ± 0.07	$\mathbf{0.94} \pm 0.01$ 0.83 ± 0.03	0.86 ± 0.05 0.79 ± 0.03
χ-Sim$_p^1$	Pr NMI	$\mathbf{0.95} \pm 0.00$	$\mathbf{0.97} \pm 0.01$	0.78 ± 0.03	$\mathbf{0.98} \pm 0.01$ 0.86 ± 0.04	$\mathbf{0.94} \pm 0.01$ 0.83 ± 0.03	0.87 ± 0.05 0.80 ± 0.02
χ-Sim$^{0.8}$	Pr NMI	$\mathbf{0.95} \pm 0.00$	$\mathbf{0.97} \pm 0.01$	0.79 ± 0.02	$\mathbf{0.98} \pm 0.01$ 0.87 ± 0.05	$\mathbf{0.94} \pm 0.01$ 0.84 ± 0.02	$\mathbf{0.90} \pm 0.01$ 0.81 ± 0.02
χ-Sim$_p^{0.8}$	Pr NMI	$\mathbf{0.95} \pm 0.00$	$\mathbf{0.97} \pm 0.01$	$\mathbf{0.80} \pm 0.04$	$\mathbf{0.98} \pm 0.00$ 0.88 ± 0.03	$\mathbf{0.94} \pm 0.01$ 0.85 ± 0.02	$\mathbf{0.90} \pm 0.02$ 0.81 ± 0.03

5. Discussion about the Preprocessing

The feature selection step aims at improving the results by removing words that are not useful to separate the different clusters of documents. Moreover, this step is also clearly needed due to the spatial and time complexity of the algorithms in $\mathcal{O}(n^3)$. Nevertheless, we are performing an *unsupervised learning* task, thus using a *supervised feature selection* method, i.e. selecting the top 2,000 words based on how much information they bring to one class of documents or another, introduces some bias since it leads to ease the problem by building well-separated clusters. In real applications, it is impossible to use this kind of preprocessing for unsupervised learning. Thus to explore the potential effects of this bias, we hereby propose to generate similar subsets of the NG20 dataset but using an *unsupervised feature selection* method.

5.1. Unsupervised Feature Selection

To reduce the number of words in the learning set, we used an approach consisting in selecting a representative subset (sampling) of the words with the help of the k-medoids algorithm. The procedure is the following: first, we remove from the corpus the words appearing in just one document, as they do not provide information to built the clusters; then, we run k-medoids to get 2,000 classes corresponding to a selection of 2,000 words. We used the implementation of the algorithm provided in the *Pycluster* package (de Hoon *et al.*, 2004) with the Euclidean distance.

5.2. Results with k-medoids

Here, we use the same methodology as described in section 4.1. except for the feature selection step which is now done with k-medoids instead of the supervised Mutual Information. The results are summarized in Table 3.

Table 3: Micro-averaged precision and standard deviation for the various subsets of NG20, pre-processed using the k-medoids feature selection.

	M2	M5	M10	NG1	NG2	NG3
Cosine	0.61 ± 0.04	0.54 ± 0.08	0.39 ± 0.03	0.52 ± 0.01	0.60 ± 0.05	0.49 ± 0.02
LSA	0.79 ± 0.09	0.66 ± 0.05	0.44 ± 0.04	0.56 ± 0.05	0.61 ± 0.06	0.52 ± 0.03
ITCC	0.70 ± 0.05	0.54 ± 0.05	0.29 ± 0.05	0.61 ± 0.06	0.44 ± 0.08	0.49 ± 0.07
SNOS	0.51 ± 0.01	0.26 ± 0.04	0.20 ± 0.02	0.51 ± 0.00	0.24 ± 0.01	0.22 ± 0.02
CTK	0.67 ± 0.10	0.76 ± 0.04	0.54 ± 0.05	0.69 ± 0.14	0.64 ± 0.06	0.54 ± 0.02
χ-Sim	0.58 ± 0.07	0.62 ± 0.12	0.43 ± 0.04	0.54 ± 0.03	0.60 ± 0.12	0.47 ± 0.05
χ-Sim$_p$	0.65 ± 0.09	0.68 ± 0.06	0.47 ± 0.04	0.62 ± 0.12	0.63 ± 0.14	0.57 ± 0.04
χ-Sim^1	0.54 ± 0.06	0.62 ± 0.13	0.36 ± 0.04	0.53 ± 0.02	0.35 ± 0.09	0.30 ± 0.05
χ-Sim$_p^1$	0.80 ± 0.13	0.77 ± 0.08	0.53 ± 0.05	0.75 ± 0.07	**0.73** ± 0.06	0.61 ± 0.03
χ-Sim$^{0.8}$	0.54 ± 0.05	0.66 ± 0.07	0.37 ± 0.06	0.52 ± 0.02	0.38 ± 0.08	0.36 ± 0.04
χ-Sim$_p^{0.8}$	**0.81** ± 0.10	**0.79** ± 0.05	**0.55** ± 0.04	**0.81** ± 0.02	0.72 ± 0.02	**0.64** ± 0.04

Here, we can observe that the results are different. The version of χ-Sim using the previous normalization method obtains more or less the same results as LSA and is totally overcome by CTK. With the new normalization the results are more contrasted and now, differently from the first experiments, the impact of the pruning factor p becomes very strong: without pruning the new method performs poorly on several problems (M2, M10, NG2, NG3), the results being lower than the Cosine similarity, but by pruning the smallest values of the similarity matrices the situation is completely opposite and χ-Sim$_p$

based algorithms obtain the best results on all the datasets. As in section 4., we observe again that setting a value of k lower than 1 improves the clustering in all the dataset but one. Now it is interesting to see with more details

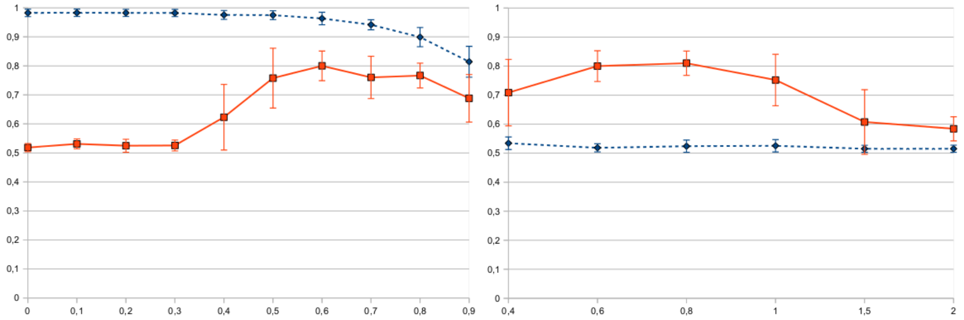

Figure 2: Evolution of the precision for NG1 using χ-Sim$_p^{0.6}$ against p (left side), and using χ-Sim$_{0.6}^k$ against k (right side). The dotted line represents the supervised feature selection data, and the plain one the unsupervised feature selection data.

the impact of the different values of k and p on a given dataset. Figure 2 (left side) shows the evolution of the accuracy on NG1 subset according to the value of p. When the words are selected by supervised mutual information the curve is quite flat, but when the words are selected with k-medoids, the behavior differs: the accuracy first increases with the pruning level, the best value being about 60% (it is worth noticing that this value is very stable among the datasets). This re-enforces our assumption that pruning the similarity matrices can be a good way of dealing with 'noise'. Indeed, when the features are selected with supervised mutual information, the classes are relatively well separated thus, similarity propagation as a result of higher order co-occurrences between documents (or words) of different categories has almost no influence. However, with the unsupervised feature selection, there are more 'noise' in the data and the pruning process helps significantly to alleviate this problem.

Figure 2 (right side) shows the evolution of the accuracy again on the NG1 subset according to the value of k. As we can see, on this dataset where the document and words vectors tend to be highly sparse, the best values for this parameter seems to be found between 0.5 and 1 as for the case of the norm L_k (Aggarwal *et al.*, 2001), we choose the value 0.8 in the results tables. However, this effect can only be seen when the pruning parameter is activated (plain line on the figure). Finally, it is worth noting that in this experiments with k-medoids, the difference between LSA and Cosine strongly decreases. All these results demonstrate that preprocessing the data with a supervised feature selection approach totally change (unsurprisingly) the behavior of the clustering methods by simplifying too much the problem.

6. Conclusion

In this paper, we proposed two empirical improvements of the χ-Sim co-similarity measure. The new normalization we presented for this measure is more consistent with the framework of the algorithm and also (partially) satisfies the reflexivity property. Furthermore, we showed that the χ-Sim similarity measure is susceptible to noise and proposed a way to alleviate this susceptibility to improve the precision. On the experimental part, our co-similarity based approach performs significantly better than the other co-clustering algorithms we tested for the task of document clustering. In contrast to Dhillon *et al.* (2003); Long *et al.* (2005), our algorithms does not need to cluster the words (columns) for clustering the documents (rows), thus avoiding the need to know the number of word clusters and the learning parameters p and k introduced here seems relatively easy to tune. However, we will investigate how to automatically find the best values for these parameters, using similarity matrix analysis from Aggarwal *et al.* (2001). It is also worth noting that our co-similarity measure performs better than LSA and than CTK by a smaller margin. Unfortunately, as we saw in section 3.1., the current method is not well-defined from the theoretical point of view and we need to analyze its behavior in order to understand the role of the pseudo-normalization and to see if it is possible to turn it into a proper normalization.

Acknowledgment

This work is partially supported by the French ANR project FRAGRANCES under grant 2008-CORD 00801.

References

AGGARWAL C. C., HINNEBURG A. & KEIM D. A. (2001). On the surprising behavior of distance metrics in high dimensional space. In *Lecture Notes in Computer Science*, p. 420–434: Springer.

BANERJEE A. & GHOSH J. (2002). Frequency sensitive competitive learning for clustering on high-dimensional hyperspheres.

BEYER K., GOLDSTEIN J., RAMAKRISHNAN R. & SHAFT U. (1999). When is "nearest neighbor" meaningful? In *Int. Conf. on Database Theory*, p. 217–235.

BISSON G. & HUSSAIN F. (2008). Chi-sim: A new similarity measure for the co-clustering task. In *Proceedings of the Seventh ICMLA*, p. 211–217: IEEE Computer Society.

CHENG Y. & CHURCH G. M. (2000). Biclustering of expression data. In *Proceedings of the International Conference on Intelligent System for Molecular Biology*, p. 93–103, Boston.

DE HOON M., IMOTO S., NOLAN J. & MIYANO S. (2004). Open source clustering software. *Bioinformatics*, p. 781.

DEERWESTER S., DUMAIS S. T., FURNAS G. W., THOMAS & HARSHMAN R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, **41**, 391–407.

DHILLON I. S., MALLELA S. & MODHA D. S. (2003). Information-theoretic co-clustering. In *Proceedings of the Ninth ACM SIGKDD*, p. 89–98.

LEMAIRE B. & DENHIÈRE G. (2008). Effects of high-order co-occurrences on word semantic similarities. *Current Psychology Letters - Behaviour, Brain and Cognition*, **18(1)**.

LIU N., ZHANG B., YAN J., YANG Q., YAN S., CHEN Z., BAI F. & YING MA W. (2004). Learning similarity measures in non-orthogonal space. In *Proceedings of the 13th ACM CIKM*, p. 334–341: ACM Press.

LIVEZAY K. & BURGESS C. (1998). Mediated priming in high-dimensional meaning space: What is "mediated" in mediated priming? In *Proceedings of the Cognitive Science Society*.

LONG B., ZHANG Z. M., WÚ X. & YU P. S. (2006). Spectral clustering for multi-type relational data. In *Procceedings of ICML'06*, p. 585–592, New York, NY, USA: ACM.

LONG B., ZHANG Z. M. & YU P. S. (2005). Co-clustering by block value decomposition. In *Proceedings of the Eleventh ACM SIGKDD*, p. 635–640, New York, NY, USA: ACM.

MADEIRA S. C. & OLIVEIRA A. L. (2004). Biclustering algorithms for biological data analysis: A survey.

QAMAR A. M. & GAUSSIER E. (2009). Online and batch learning of generalized cosine similarities. In *Proceedings of the Ninth IEEE ICDM*, p. 926–931, Washington, DC, USA: IEEE Computer Society.

REGE M., DONG M. & FOTOUHI F. (2008). Bipartite isoperimetric graph partitioning for data co-clustering. *Data Min. Knowl. Discov.*, **16**(3), 276–312.

SALTON G. (1971). *The SMART Retrieval System—Experiments in Automatic Document Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

SENETA E. (2006). *Non-Negative Matrices and Markov Chains*. Springer.

SLONIM N. & TISHBY N. (2001). The power of word clusters for text classification. In *In 23rd European Colloquium on Information Retrieval Research*.

SPEER N., SPIETH C. & ZELL A. (2004). A memetic clustering algorithm for the functional partition of genes based on the gene ontology.

WANG X.-J., MA W.-Y., XUE G.-R. & LI X. (2004). Multi-model similarity propagation and its application for web image retrieval. In *Proceedings of the 12th annual ACM MULTIMEDIA*, p. 944–951, New York, NY, USA: ACM.

YANG Y. & PEDERSEN J. O. (1997). A comparative study on feature selection in text categorization. In *ICML*, p. 412–420.

YEN L., FOUSS F., DECAESTECKER C., FRANCQ P. & SAERENS M. (2009). Graph nodes clustering with the sigmoid commute-time kernel: A comparative study. *Data Knowl. Eng.*, **68**(3), 338–361.

ZELIKOVITZ S. & HIRSH H. (2001). Using lsi for text classification in the presence of background text. In *Proceedings of the 10th ACM CIKM*, p. 113–118: ACM Press.

ZHANG D., ZHOU Z.-H. & CHEN S. (2007). Semi-supervised dimensionality reduction. In *Proceedings of the SIAM ICDM*.