# Hypermetric k-Means Clustering for Content-based Document Management

Sergio Decherchi, Paolo Gastaldo, Judith Redi, Rodolfo Zunino,

Dept. Biophysical and Electronic Engineering, University of Genoa,
16145 Genova, Italy
{sergio.decherchi, paolo.gastaldo, judith.redi, rodolfo.zunino}@unige.it

**Abstract.** Text-mining methods have become a key feature for homeland-security technologies, as they can help explore effectively increasing masses of digital documents in the search for relevant information. This research presents a model for document clustering that arranges unstructured documents into content-based homogeneous groups. The overall paradigm is hybrid because it combines pattern-recognition grouping algorithms with semantic-driven processing. First, a semantic-based metric measures distances between documents, by combining a content-based with a behavioral analysis; the metric considers both lexical properties and the structure and styles that characterize the processed documents. Secondly, the model relies on a Radial Basis Function (RBF) kernel-based mapping for clustering. As a result, the major novelty aspect of the proposed approach is to exploit the implicit mapping of RBF kernel functions to tackle the crucial task of normalizing similarities while embedding semantic information in the whole mechanism.

**Keywords:** document clustering, homeland security, kernel k-means.

## 1 Introduction

The automated surveillance of information sources is of strategic importance to effective homeland security [1,2]. The increased availability of data-intensive heterogeneous sources provides a valuable asset for the intelligence task; data-mining methods have therefore become a key feature for security-related technologies [2,3] as they can help explore effectively increasing masses of digital data in the search for relevant information.

Text mining techniques provide a powerful tool to deal with large amounts of unstructured text data [4,5] that are gathered from any multimedia source (e.g. from Optical Character Recognition, from audio via speech transcription, from web-crawling agents, etc.). The general area of text-mining methods comprises various approaches [5]: detection/tracking tools continuously monitor specific topics over time; document classifiers label individual files and build up models for possible subjects of interest; clustering tools process documents for detecting relevant relations among those subjects. As a result, text mining can profitably support intelligence and security activities in identifying, tracking, extracting, classifying and discovering patterns, so that the outcomes can generate alerts notifications accordingly [6,7].

This work addresses document clustering and presents a dynamic, adaptive clustering model to arrange unstructured documents into content-based homogeneous groups. The framework implements a hybrid paradigm, which combines a content-driven similarity processing with pattern-recognition grouping algorithms. Distances between documents are worked out by a semantic-based hypermetric: the specific approach integrates a content-based with a user-behavioral analysis, as it takes into account both lexical and style-related features of the documents at hand. The core clustering strategy exploits a kernel-based version of the conventional k-means algorithm [8]; the present implementation relies on a Radial Basis Function (RBF) kernel-based mapping [9]. The advantage of using such a kernel consists in supporting normalization implicitly; normalization is a critical issue in most text-mining applications, and prevents that extensive properties of documents (such as length, lexicon, etc) may distort representation and affect performance.

A standard benchmark for content-based document management, the Reuters database [10], provided the experimental domain for the proposed methodology. The research shows that the document clustering framework based on kernel k-means can generate consistent structures for information access and retrieval.

## 2 Document Clustering

Text mining can effectively support the strategic surveillance of information sources thanks to automatic means, which is of paramount importance to homeland security [6,7]. For prevention, text mining techniques can help identify novel "information trends" revealing new scenarios and threats to be monitored; for investigation, these technologies can help distil relevant information about known scenarios. Within the text mining framework, this work addresses document clustering, which is one of the most effective techniques to organize documents in an unsupervised manner.

When applied to text mining, clustering algorithms are designed to discover groups in the set of documents such that the documents within a group are more similar to one another than to documents of other groups. As apposed to text categorization [5], in which categories are predefined and are part of the input information to the learning procedure, document clustering follows an unsupervised paradigm and partitions a set of documents into several subsets. Thus, the document clustering problem can be defined as follows. One should first define a set of documents $\mathcal{D} = \{D_1, . . . , D_n\}$, a similarity measure (or distance metric), and a partitioning criterion, which is usually implemented by a cost function. In the case of flat clustering, one sets the desired number of clusters, $Z$, and the goal is to compute a membership function $\phi : \mathcal{D} \rightarrow \{1, . . . , Z\}$ such that $\phi$ minimizes the partitioning cost with respect to the similarities among documents. Conversely, hierarchical clustering does not need to define the cardinality, $Z$, and applies a series of nested partitioning tasks which eventually yield a hierarchy of clusters.

Indeed, every text mining framework should always be supported by an information extraction (IE) model [11,12] which is designed to pre-process digital text documents and to organize the information according to a given structure that can be directly interpreted by a machine learning system. Thus, a document $D$ is eventually reduced to a sequence of terms and is represented as a vector, which lies in a space spanned by the dictionary (or vocabulary) $\mathcal{T} = \{t_j;\, j= 1,.., n_T\}$. The dictionary collects all terms used to represent any document $D$, and can be assembled empirically by gathering the terms that occurs at least once in a document collection $\mathcal{D}$ ; by this representation one loses the original relative ordering of terms within each document. Different models [11,12] can be used to retrieve index terms and to generate the vector that represents a document $D$. However, the vector space model [13] is the most widely used method for document clustering. Given a collection of documents $\mathcal{D}$, the vector space model represents each document $D$ as a vector of real-valued weight terms $\mathbf{v} = \{w_j;\, j=1,..,n_T\}$. Each component of the $n_T$-dimensional vector is a non-negative term weight, $w_j$, that characterizes the $j$-th term and denotes the relevance of the term itself within the document $D$.

# 3 Hypermetric k-means clustering

The hybrid approach described in this Section combines the specific advantages of content-driven processing with the effectiveness of an established pattern-recognition grouping algorithm. Document similarity is defined by a content-based distance, which combines a classical distribution-based measure with a behavioral analysis of the style features of the compared documents. The core engine relies on a kernel-based version of the classical k-means partitioning algorithm [8] and groups similar documents by a top-down hierarchical process. In the kernel-based approach, every document is mapped into an infinite-dimensional Hilbert space, where only inner products among elements are meaningful and computable. In the present case the kernel-based version of k-means [15] provides a major advantage over the standard k-means formulation.

In the following, $\mathcal{D} = \{D_u;\, u= 1,..,n_D\}$ will denote the corpus, holding the collection of documents to be clustered. The set $\mathcal{T} = \{t_j;\, j= 1,.., n_T\}$ will denote the vocabulary, which is the collection of terms that occur at least one time in $\mathcal{D}$ after the pre-processing steps of each document $D \in \mathcal{D}$ (e.g., stop-words removal, stemming [11]).

## 3.1 Document distance measure

A novel aspect of the method described here is the use of a document-distance that takes into account both a conventional content-based similarity metric and a behavioral similarity criterion. The latter term aims to improve the overall performance of the clustering framework by including the structure and style of the documents in the process of similarity evaluation. To support the proposed document distance measure, a document $D$ is here represented by a pair of vectors, $\mathbf{v}'$ and $\mathbf{v}''$.

Vector $\mathbf{v}'(D)$ actually addresses the content description of a document $D$; it can be viewed as the conventional $n_T$-dimensional vector that associates each term $t \in \mathcal{T}$

with the normalized frequency, *tf*, of that term in the document *D*. Therefore, the *k*-th element of the vector **v**′($D_u$) is defined as:

$$v'_{k,u} = tf_{k,u} \Big/ \sum_{l=1}^{n_T} tf_{l,u} \quad , \tag{1}$$

where $tf_{k,u}$ is the frequency of the *k*-th term in document $D_u$. Thus **v**′ represents a document by a classical vector model, and uses term frequencies to set the weights associated to each element.

From a different perspective, the structural properties of a document, *D*, are represented by a set of probability distributions associated with the terms in the vocabulary. Each term $t \in \mathcal{T}$ that occurs in $D_u$ is associated with a distribution function that gives the spatial probability density function (pdf) of *t* in $D_u$. Such a distribution, $p_{t,u}(s)$, is generated under the hypothesis that, when detecting the *k*-th occurrence of a term *t* at the normalized position $s_k \in [0,1]$ in the text, the spatial pdf of the term can be approximated by a Gaussian distribution centered around $s_k$. In other words, if the term $t_j$ is found at position $s_k$ within a document, another document with a similar structure is expected to include the same term at the same position or in a neighborhood thereof, with a probability defined by a Gaussian pdf. To derive a formal expression of the pdf, assume that the *u*-th document, $D_u$, holds $n_O$ occurrences of terms after simplifications; if a term occurs more than once, each occurrence is counted individually when computing $n_O$, which can be viewed as a measure of the length of the document. The spatial pdf can be defined as:

$$p_{t,u}(s) = \frac{1}{A}\sum_{k=1}^{n_O} G(s_k,\lambda) = \frac{1}{A}\sum_{k=1}^{n_O}\frac{1}{\sqrt{2\pi}\lambda}\exp\left[-\frac{(s-s_k)^2}{\lambda^2}\right] \quad , \tag{2}$$

where *A* is a normalization term and λ is regularization parameter. In practice one uses a discrete approximation of (2). First, the document *D* is segmented evenly into *S* sections. Then, an *S*-dimensional vector is generated for each term $t \in \mathcal{T}$, and each element estimates the probability that the term *t* occurs in the corresponding section of the document. As a result, **v**″(*D*) is an array of $n_T$ vectors having dimension *S*.

Vector **v**′ and vector **v**″ support the computations of the frequency-based distance, Δ(*f*), and the behavioral distance, Δ(*b*), respectively. The former term is usually measured according to a standard Minkowski distance, hence the content distance between a pair of documents ($D_u$, $D_v$) is defined by:

$$\Delta^{(f)}(D_u, D_v) = \left[\sum_{k=1}^{n_T}\left|v'_{k,u} - v'_{k,v}\right|^p\right]^{1/p} \quad . \tag{3}$$

The present approach adopts the value $p = 1$ and therefore actually implements a Manhattan distance metric. The term computing behavioral distance, Δ(*b*), applies an Euclidean metric to compute the distance between probability vectors **v**″. Thus:

$$\Delta^{(b)}(D_u, D_v) = \sum_{k=1}^{n_T}\Delta^{(b)}_{t_k}(D_u, D_v) = \sum_{k=1}^{n_T}\sum_{s=1}^{S}\left[v''_{(k)s,u} - v''_{(k)s,v}\right]^2 \quad . \tag{4}$$

Both terms (3) and (4) contribute to the computation of the eventual distance value, $\Delta(D_u, D_v)$, which is defined as follows:

$$\Delta(D_u,D_v) = \alpha\cdot \Delta^{(f)} (D_u,D_v) + (1 - \alpha)\cdot \Delta^{(b)}(D_u,D_v) \ , \tag{5}$$

where the mixing coefficient $\alpha\in[0,1]$ weights the relative contribution of $\Delta(f)$ and $\Delta(b)$. It is worth noting that the distance expression (5) obeys the basic properties of non-negative values and symmetry that characterize general metrics, but does not necessarily satisfy the triangular property.

### 3.2 Kernel k-means

The conventional k-means paradigm supports an unsupervised grouping process [8], which partitions the set of samples, $\mathcal{D} = \{D_u;\ u= 1,..,n_D\}$, into a set of $Z$ clusters, $C_j$ ($j$ = 1,…, $Z$). In practice, one defines a "membership vector," which indexes the partitioning of input patterns over the $K$ clusters as: $m_u = j \Leftrightarrow D_u \in \mathrm{C}_j$, otherwise $m_u$ = 0; $u$ = 1,…, $n_D$. It is also useful to define a "membership function" $\delta_{uj}(D_u,\mathrm{C}_j)$, that defines the membership of the $u$-th document to the $j$-th cluster: $\delta_{uj} =1$ if $m_u = j$, and 0 otherwise. Hence, the number of members of a cluster is expressed as

$$N_j = \sum_{u=1}^{n_D} \delta_{uj} \ ; \quad j = 1,\ldots, Z\ ; \tag{6}$$

and the cluster centroid is given by:

$$\mathbf{w}_j = \frac{1}{N_j}\sum_{u=1}^{n_D} \mathbf{x}_u \delta_{uj} \ ; \quad j = 1,\ldots, Z\ ; \tag{7}$$

where $\mathbf{x}_u$ is any vector-based representation of document $D_u$.

The kernel based version of the algorithm is based on the assumption that a function, $\Phi$, can map any element, $D$, into a corresponding position, $\Phi(D)$, in a possibly infinite dimensional Hilbert space. The mapping function defines the actual 'Kernel', which is formulated as the expression to compute the inner product:

$$K(D_u,D_v) = K_{uv} = \Phi(D_u)\cdot \Phi(D_v) \overset{def}{=} \Phi_u \cdot \Phi_v \ . \tag{8}$$

In our particular case we employ the largely used RBF kernel

$$K(D_u, D_v) = \exp\left[-\frac{\Delta(D_u, D_v)}{\sigma^2}\right] . \tag{9}$$

It is worth stressing here an additional, crucial advantage of using a kernel-based formulation in the text-mining context: the approach (9) can effectively support the critical normalization process by reducing all inner products within a limited range, thereby preventing that extensive properties of documents (length, lexicon, etc) may distort representation and ultimately affect clustering performance. The kernel-based version of the k-means algorithm, according to the method proposed in [15], replicates the basic partitioning schema (6)-(7) in the Hilbert space, where the centroid positions, $\Psi$, are given by the averages of the mapping images, $\Phi_u$:

$$\Psi_j = \frac{1}{N_j}\sum_{u=1}^{n_D}\Phi_u\delta_{uj} \quad ; \quad j = 1,\ldots,Z . \tag{10}$$

The ultimate result of the clustering process is the membership vector, **m**, which determines prototype positions (7) even though they cannot be stated explicitly. As a consequence, for a document, $D_u$, the distance in the Hilbert space from the mapped image, $\Phi_u$, to the cluster $\Psi_j$ as per (7) can be worked out as:

$$d\left(\Phi_u, \Psi_j\right) = \left\| \Phi_u - \frac{1}{N_j}\sum_{v=1}^{n_D}\Phi_v \right\|^2 = 1 + \frac{1}{\left(N_j\right)^2}\sum_{m,v=1}^{n_D}\delta_{mj}\delta_{vj}K_{mv} - \frac{2}{N_j}\sum_{v=1}^{n_D}\delta_{vj}K_{u,v} \ . \tag{11}$$

By using expression (11), which includes only kernel computations, one can identify the closest prototype to the image of each input pattern, and assign sample memberships accordingly.

In clustering domains, k-means clustering can notably help separate groups and discover clusters that would have been difficult to identify in the base space. From this viewpoint one might even conclude that a kernel-based method might represent a viable approach to tackle the dimensionality issue.

## 4 Experimental Results

A standard benchmark for content-based document management, the Reuters database [10], provided the experimental domain for the proposed framework. The database includes 21,578 documents, which appeared on the Reuters newswire in 1987. One or more topics derived from economic subject categories have been associated by human indexing to each document; eventually, 135 different topics were used. In this work, the experimental session involved a corpus $\mathcal{D}_R$ including 8267 documents out of the 21,578 originally provided by the database. The corpus $\mathcal{D}_R$ was obtained by adopting the criterion used in [14]. First, all the documents with multiple topics were discarded. Then, only the documents associated to topics having at least 18 occurrences were included in $\mathcal{D}_R$. As a result, 32 topics were represented in the corpus.

In the following experiments, the performances of the clustering framework have been evaluated by using the purity parameter. Let $N_k$ denote the number of elements lying in a cluster $C_k$ and let $N_{mk}$ be the number of elements of the class $I_m$ in the cluster $C_k$. Then, the purity $pur(k)$ of the cluster $C_k$ is defined as follows:

$$pur(k) = \frac{1}{N_k}\max_m(N_{mk}) \ . \tag{12}$$

Accordingly, the overall purity of the clustering results is defined as follows:

$$purity = \sum_k \frac{N_k}{N} \cdot pur(k) \ , \tag{13}$$

where $N$ is the total number of element. The purity parameter has been preferred to other measures of performance (e.g. the F-measures) since it is the most accepted measure for machine learning classification problems [11].

The clustering performance of the proposed methodology was evaluated by analyzing the result obtained with three different experiments: the documents in the corpus $\mathcal{D}_R$ were partitioned by using a flat clustering paradigm and three different settings for the parameter $\alpha$, which, as per (5), weights the relative contribution of $\Delta(f)$ and $\Delta(b)$ in the document distance measure. The values used in the experiments were $\alpha = 0.3$, $\alpha = 0.7$ and $\alpha = 0.5$; thus, a couple of experiments were characterized by a strong preponderance of one of the two components, while in the third experiment $\Delta(f)$ and $\Delta(b)$ evenly contribute to the eventual distance measure.

Table 1 outlines the results obtained with the setting $\alpha = 0.3$. The evaluations were conducted with different number of clusters $Z$, ranging from 20 to 100. For each experiment, four quality parameters are presented:

- the overall purity, $purity_{OV}$, of the clustering result;
- the lowest purity value $pur(k)$ over the $Z$ clusters;
- the highest purity value $pur(k)$ over the $Z$ clusters;
- the number of elements (i.e. documents) associated to the smallest cluster.

Analogously, Tables 2 and 3 reports the results obtained with $\alpha = 0.5$ and $\alpha = 0.7$, respectively.

As expected, the numerical figures show that, in general, the overall purity grows as the number of clusters $Z$ increases. Indeed, the value of the overall purity seems to indicate that clustering performances improve by using the setting $\alpha = 0.3$. Hence, empirical outcomes confirm the effectiveness of the proposed document distance measure, which combines the conventional content-based similarity with the behavioral similarity criterion.

**Table 1.** Clustering performances obtained on Reuters-21578 with $\alpha$=0.3.

| Number of clusters | Overall purity | *pur*(*k*) minimum | *pur*(*k*) maximum | Smallest cluster |
|---|---|---|---|---|
| 20 | 0.712108 | 0.252049 | 1 | 109 |
| 40 | 0.77138 | 0.236264 | 1 | 59 |
| 60 | 0.81154 | 0.175 | 1 | 13 |
| 80 | 0.799685 | 0.181818 | 1 | 2 |
| 100 | 0.82666 | 0.153846 | 1 | 1 |

**Table 2.** Clustering performances obtained on Reuters-21578 with $\alpha$=0.5.

| Number of clusters | Overall purity | *pur*(*k*) minimum | *pur*(*k*) maximum | Smallest cluster |
|---|---|---|---|---|
| 20 | 0.696383 | 0.148148 | 1 | 59 |
| 40 | 0.782267 | 0.222467 | 1 | 4 |
| 60 | 0.809121 | 0.181818 | 1 | 1 |
| 80 | 0.817467 | 0.158333 | 1 | 1 |
| 100 | 0.817467 | 0.139241 | 1 | 2 |

**Table 3.** Clustering performances obtained on Reuters-21578 with $\alpha$=0.7.

| Number of clusters | Overall purity | *pur*(*k*) minimum | *pur*(*k*) maximum | Smallest cluster |
|---|---|---|---|---|
| 20 | 0.690577 | 0.145719 | 1 | 13 |
| 40 | 0.742833 | 0.172638 | 1 | 6 |
| 60 | 0.798718 | 0.18 | 1 | 5 |
| 80 | 0.809483 | 0.189655 | 1 | 2 |
| 100 | 0.802589 | 0.141732 | 1 | 4 |

## References

1. Chen, H., Chung, W., Xu, J.J., Wang, G., Qin, Y., Chau, M.: Crime data mining: a general framework and some examples. IEEE Trans. Computer. 37, 50--56 (2004)
2. Seifert, J. W.: Data Mining and Homeland Security: An Overview. CRS Report RL31798, www.epic.org/privacy/fusion/crs-dataminingrpt.pdf (2007)
3. Mena, J.: Investigative Data Mining for Security and Criminal Detection. Butterworth-Heinemann (2003)
4. Sullivan, D.: Document warehousing and text mining. John Wiley and Sons (2001)
5. Fan, W., Wallace, L., Rich, S., Zhang, Z.: Tapping the power of text mining. Comm. of the ACM. 49, 76--82 (2006)
6. Popp, R., Armour, T., Senator, T., Numrych, K.: Countering terrorism through information technology. Comm. of the ACM. 47, 36--43 (2004)
7. Zanasi, A. (eds.): Text Mining and its Applications to Intelligence, CRM and KM. 2nd edition , WIT Press (2007).
8. Linde, Y., Buzo, A., Gray, R.M.: An algorithm for vector quantizer design. IEEE Trans. Commun. COM-28, 84--95 (1980).
9. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge (2004).
10. Reuters-21578 Text Categorization Collection. UCI KDD Archive.
11. Manning, C. D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)
12. Baeza-Yates, R., Ribiero-Neto, B.: Modern Information Retrieval. ACM Press (1999).
13. Salton, G., Wong, A., Yang, L.S.: A vector space model for information retrieval. Journal Amer. Soc. Inform. Sci. 18, 613—620 (1975)
14. Cai, D., He, X., Han, J.: Document Clustering Using Locality Preserving Indexing. IEEE Transaction on knowledge and data engineering. 17, 1624--1637 (2005).
15. Girolami, M.: Mercer kernel based clustering in feature space. IEEE Trans. Neural Networks. 13, 2780--2784 (2002).