Data & Knowledge Engineering 70 (2011) 284–306

Contents lists available at ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

# Document clustering using synthetic cluster prototypes

Argyris Kalogeratos, Aristidis Likas *

*Department of Computer Science, University of Ioannina, 45110, Ioannina, Greece*

ARTICLE INFO

*Article history:*
Received 17 December 2009
Received in revised form 11 December 2010
Accepted 13 December 2010
Available online 24 December 2010

*Keywords:*
Clustering methods
Document clustering
Text mining
Term selection
Subspace clustering

ABSTRACT

The use of centroids as prototypes for clustering text documents with the k-means family of methods is not always the best choice for representing text clusters due to the high dimensionality, sparsity, and low quality of text data. Especially for the cases where we seek clusters with small number of objects, the use of centroids may lead to poor solutions near the bad initial conditions. To overcome this problem, we propose the idea of *synthetic cluster prototype* that is computed by first selecting a subset of cluster objects (instances), then computing the representative of these objects and finally selecting important features. In this spirit, we introduce the *MedoidKNN* synthetic prototype that favors the representation of the *dominant class* in a cluster. These synthetic cluster prototypes are incorporated into the generic spherical k-means procedure leading to a robust clustering method called *k-synthetic prototypes* (*k-sp*). Comparative experimental evaluation demonstrates the robustness of the approach especially for small datasets and clusters overlapping in many dimensions and its superior performance against traditional and subspace clustering methods.



## 1. Introduction

Document clustering is an unsupervised learning approach for automatically segregating similar documents of a corpus into the same group, called *cluster*, and dissimilar documents to different groups. Formally, a corpus of $N$ unlabeled documents is given and a solution $C=\{c_j: j=1,\ldots, k\}$ is searched that partitions the document into $k$ disjoint clusters.

Even small text datasets carry large vocabularies and certain undesirable effects arise due to the *curse of dimensionality* [4]. The *high dimensional and sparse* (*HDS*) feature space in combination with language phenomena such as *polysemy*, *homosemy* and *metaphors*, constitute an adverse setting for clustering methods. When a labeled training dataset is provided, several statistical options are available for feature selection [5,6], even in case of multilabeled data objects [53]. On the other hand, it is more complicated to select features in an unsupervised setting and it is usually achieved using heuristics [49–52]. Methods such as *Latent Semantic Indexing* (*LSI*) [47], or *Latent Dirichlet Allocation* [48] (*LDA*), may discover the term correlations but they map the data into a feature space of much lower dimensionality.

Clustering algorithms are separated in two major categories *hierarchical* and *partitional* (for a survey see [7]). The former produce a hierarchy of solutions, either by merging, or by dividing clusters. Partitional approaches seek to discover a set of unique cluster representations that describe properly the underlying data classes of a dataset. An *objective function* $\Phi(C)$ evaluates the quality of a data partition by quantifying how good the derived representations are for the corresponding clusters. These methods start from a set of $k$ cluster representations which are improved iteratively in a way that $\Phi(C)$ is optimized. *Probabilistic* methods use probabilistic *cluster models* (or *topic models*) [8,9], while *non-probabilistic* methods utilize representatives in the feature space,

* Corresponding author. Tel.: +30 26510 08810; fax: +30 26510 08882.
*E-mail addresses:* akaloger@cs.uoi.gr (A. Kalogeratos), arly@cs.uoi.gr (A. Likas).

0169-023X/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.datak.2010.12.002

called *prototypes*, that are used to represent the objects of a cluster. Typical prototypes are the *arithmetic mean* called *centroid*, and the *medoid* that is a real object which is representative for the cluster it belongs.

A popular partitional method is *k-means* [10] that represents each cluster with its centroid. Many heuristic variations of k-means have been proposed and applied for text collections [11–14]. *Spherical k-means* (*spk-means*) [22] is a modified version that utilizes the *cosine similarity* measure to cluster the data by partitioning the unit hypersphere into $k$ hypercones, one for each cluster. This method is fast and gives better clusters than traditional k-means [12].

Special algorithms have also been developed to deal with HDS feature spaces. The clustering methodology aiming at finding clusters in subspaces of data instead of the entire feature space is referred to as *subspace clustering* and its key characteristic is the simultaneous determination of the object membership to clusters and the subspace of each cluster. Surveys on subspace clustering in high dimensional spaces can be found in [29,46]. Recently, much attention has been received by methods that aim to identify the cluster structure in on-line high dimensional data streams [54,55].

This work puts forth the idea that, although the centroids are the optimal cluster prototypes with respect to certain objective functions (e.g. based on cosine similarity), their optimality could also become a drawback in HDS feature spaces and in cases of low data quality (e.g. outliers, noise). Especially, as the number of data objects becomes smaller compared to the complexity of a clustering problem (i.e. number of clusters, dimensionality), the centroids become less appropriate cluster representatives. Text documents constitute a typical example of data where such an adverse setting is met.

In this paper we present the *synthetic prototype*, a novel type of cluster representative that, given the object assignment to clusters, is computed in two steps: a) a *reference prototype* is constructed for the cluster and then b) *feature selection* is applied on it. We propose the so-called *MedoidKNN* reference prototype which is based on a subset of $K$ objects of a cluster that are close to its medoid. This synthetic prototype favors the representation of the objects of the *dominant class* in a cluster, i.e. the class to which the majority of the cluster objects belong. Finally, we modify the generic spk-means iterative procedure by incorporating synthetic prototypes. This leads to a novel, effective and quite simple clustering method called *k-synthetic prototypes* (*k-sp*). We conducted an extensive evaluation of the k-sp method examining several options for the synthetic prototypes and comparing it to several traditional clustering methods such as spherical k-means, agglomerative, spectral clustering and two soft subspace clustering methods.

The rest of this paper is organized as follows: in Section 2, a background discussion for the document clustering problem is provided. In Section 3, we present our novel synthetic prototype cluster representation and the k-synthetic prototypes clustering method. In Section 4, comparative experimental results are reported and discussed, and finally in Section 5, we present concluding remarks and future research directions.

## 2. Background

### 2.1. Document representation

A preprocessing step on the corpus decides which terms are meaningful to be included in the *corpus vocabulary* $V$, a set of $|V|$ unique features. Despite the fact that it is reasonable to seek for complex representations for text data, such as graphs [1–3], the typical approach is to represent each input document as a *bag-of-words* [18] feature vector $d_i \in \mathbb{R}^{|V|}$, $i=1,\ldots,N$, whose elements are weight values denoting the significance of each vocabulary term for the document. Typically, the weights are set using the $tf \times idf$ scheme and document vectors are normalized to unit length with respect to Euclidean $L_2$-norm. Hence, the $i$-th document is modeled as:

$$d_i = \frac{\left(tf_{i1}\log\frac{N}{N^{(1)}},\ldots,tf_{i|V|}\log\frac{N}{N^{(|V|)}}\right)}{\left[\sum_{j=1}^{|V|} tf_{ij}^2\log^2\frac{N}{N^{(j)}}\right]^{-1/2}}, \tag{1}$$

where $tf_{ij}$ is the frequency of $j$-th term in the $i$-th document and $N^{(j)}$ the number of documents that contain $j$-th term. The proximity between two documents is computed using *cosine similarity*, considered to be an effective measure for text clustering [19,20], that computes the cosine of the angle between the two document vectors:

$$sim^{(cos)}\left(d_i,d_j\right) = \cos\left(\theta\left(d_i,d_j\right)\right) = \frac{d_i^{\top}}{\|d_i\|_2}\cdot\frac{d_j}{\|d_j\|_2}. \tag{2}$$

### 2.2. Properties of the representation space of documents

The properties of the vector space in which text documents are represented are closely related to the nature of human language. Even small text datasets carry very large vocabularies and, apart from the known negative effects of the curse of dimensionality, the learning algorithms have to deal with the existence of high sparsity. It has been observed that a document may have less than 1% of the global corpus vocabulary [21] (non-zero vector dimensions) since there are terms in the corpus vocabulary that do not appear in a given document although they are relevant to its content. This is due to the fact that each document usually is a specific *semantically narrow instance* of a much more general document class.