Clustering Objects from Multiple Collections*

Vera Hollink**, Maarten van Someren, and Viktor de Boer

University of Amsterdam

Abstract. Clustering methods cluster objects on the basis of a similarity measure between the objects. In clustering tasks where the objects come from more than one collection often part of the similarity results from features that are related to the collections rather than features that are relevant for the clustering task. For example, when clustering pages from various web sites by topic, pages from the same web site often contain similar terms. The collection-related part of the similarity hinders clustering as it causes the creation of clusters that correspond to collections instead of topics. In this paper we present two methods to restrict clustering to the part of the similarity that is not associated with membership of a collection. Both methods can be used on top of standard clustering methods. Experiments on data sets with objects from multiple collections show that our methods result in better clusters than methods that do not take collection information into account.

1 Introduction

In many clustering applications we have an a priori idea about the types of clusters we are looking for. For example, in document clustering tasks we often want clusters that correspond to the documents' topics. Even though the cluster type is known, clustering is still unsupervised, as we do not know in advance *which* topics are present in the data. Clustering algorithms form clusters of objects that are similar to each other in terms of some measure of similarity. For homogeneous sets of objects standard similarity or distance measures usually lead to satisfying results. For instance, cosine distance applied to word vectors is suitable for finding topic clusters in a news archive or web site (e.g. [1,2]).

When the data comes from multiple collections, often the collections do not coincide with the type of clusters that we want to find. In this situation we know in advance that features that are related to the collections are not relevant for our clustering task. The part of the similarity which is associated with these features can hinder clustering as it causes clustering algorithms to group objects primarily by collection. For example, this problem occurs when we want to cluster pages from a number of web sites by topic using a word-based similarity measure. In terms of such a measure pages from the same web sites are usually more similar to each other than to pages from other web sites, because pages from one web

* This research is part of the project 'Adaptive generation of workflow models for human-computer interaction' (project MMI06101) funded by SenterNovem.

** Currently at Centre for Mathematics and Computer Science (CWI).

B. Mertsching, M. Hund, and Z. Aziz (Eds.): KI 2009, LNAI 5803, pp. 136–143, 2009.

site share a common terminology and often contain the same names. As a result, we get clusters of pages from the same site instead of clusters of pages on the same topic. Similarity, with image clustering, images can become clustered by illumination or background instead by the things that they depict.

The task of clustering objects from multiple collections can be formalized as follows. There is a set of objects $\mathcal{I}$ and a set of collections $\mathcal{R}$. The collection of each object is defined by a special feature $col : \mathcal{I} \rightarrow \mathcal{R}$. The value of *col* influences the value of an unknown part of the other features. There is a similarity function $sim : \mathcal{I} \times \mathcal{I} \rightarrow \mathbb{R}$, which comprises both relevant similarity and similarity associated with the value of *col*. The task is to divide $\mathcal{I}$ in a set of clusters that represent only relevant similarity. In this paper we do not consider overlapping or hierarchical clusters, but the proposed methods can be applied without modification to algorithms that create these types of clusters.

An intuitive solution for the problems caused by collections is to use only features that are relevant for the clustering task. Selecting the right features directly requires much effort and a deep understanding of the domain. E.g., specialized image analysis techniques are needed to compensate for illumination, background, etc. In this paper we provide two simple and generally applicable methods to deal with collection-related similarity. The methods do not require any domain knowledge, but only need to know which objects are from which collections. For a user, this is easy to determine, as the collections represent some obviously irrelevant feature such as where or by who the objects were created. The proposed methods can be used as an addition to standard clustering algorithms.

2 Related Work

Li and Liu [3] present a method for binary text classification on data sets where the training and the test set come from different web sites and thus have different term distributions. To compensate for this they add irrelevant documents as extra negative examples. This task is related to ours, but it is fully supervised.

Huang and Mitchell [4] address document clustering. They use an extension of EM to determine which terms are cluster-specific and which terms are shared by all clusters. This task differs from the one addressed in this work, in that they reinforce features of clusters that are found automatically instead of features of user-defined collections. If their method initially groups objects by collection, in later iterations it will assign even higher weights to collection-specific features.

Bronstein et al. [5] defined the concept of 'partial similarity'. Two images are partially similar when parts of the objects are similar while other parts differ, e.g. a picture of a horse and a picture of a centaur. The authors developed a method to identify a subset of the object features that correspond to the similar parts of the images. Unlike partial similarity the methods that we present are also applicable in situations in which there is no clear distinction between irrelevant and relevant features. This happens, for example, if we have a set of pictures taken with different lightings. Illumination influences the whole pictures and thus can not be compensated for by focussing on a part of the pixels.

3 Two Methods to Deal with Multiple Collections

3.1 The Omission Method

Similarity-based clustering algorithms compute the similarities between objects from the object features. During clustering these algorithms use only the similarities and not the raw features. The quality of a potential clustering $\mathcal{C}$ is computed as a function f of the similarities between objects (the objective function):

$$quality(\mathcal{C}) = f(\mathcal{C}, \{sim(i,j)|i,j \in \mathcal{I}\}) \tag{1}$$

For example, the average linkage algorithm [6] expresses the quality of a cluster as the average similarity between all pairs of objects in the cluster.

The omission method computes the same function, but omits the similarity of pairs of objects from the same collection, because these similarities often consist for a large part of collection-related similarity:

$$quality(\mathcal{C}) = f(\mathcal{C}, \{sim(i,j)|i,j \in \mathcal{I} \wedge col(i) \neq col(j)\}) \tag{2}$$

3.2 The Estimation Method

The estimation method estimates which part of the similarity between objects is relevant similarity and which part is caused by collections. The relevant part of the similarities is given to the clustering algorithm to cluster the objects.

We decompose the similarity between any two objects a and b into relevant similarity, $sim_{rel}(a,b)$, and collection-related similarity, $sim_{col}(a,b)$:

$$sim(a,b) = sim_{rel}(a,b) + sim_{col}(a,b) \tag{3}$$

We estimate $sim_{col}(a,b)$ as the average collection-related similarity between all pairs of objects in the collections of a and b. Therefore, the collection-related similarity between objects a and b from collections A and B can be expressed as the similarity between their collections, $sim_{col}(A,B)$:

$$\forall a \in A, b \in B : sim_{col}(a,b) = Avg_{\{a' \in A, b' \in B\}} sim_{col}(a',b') = sim_{col}(A,B) \tag{4}$$

When the target clusters (the clusters that we are trying to find) are independent of the collections, the target clusters will be more or less evenly spread over the collections. As a result, the average relevant similarity between objects from a pair of collections will be roughly the same for all pairs of collections. Consequently, we can estimate the average relevant similarity between objects from all pairs of collections as a constant φ:

$$\forall A, B \in \mathcal{R} : Avg_{\{a \in A, b \in B\}} sim_{rel}(a,b) = \varphi \tag{5}$$

The average similarity between the objects in a pair of collections A and B can now be expressed as:

$$\begin{aligned} Avg_{\{a \in A, b \in B\}} sim(a,b) &= Avg_{\{a \in A, b \in B\}} (sim_{rel}(a,b) + sim_{col}(a,b)) \\ &= Avg_{\{a \in A, b \in B\}} (sim_{rel}(a,b) + sim_{col}(A,B)) \\ &= \varphi + sim_{col}(A,B) \end{aligned} \tag{6}$$

The value of $Avg_{\{a \in A, b \in B\}} sim(a, b)$ can be computed directly. The correct value of φ is unknown. We define the collection-related similarity between the two most dissimilar collections to be 0 (other values lead to different absolute values for the relevant similarities, but do not change the differences between the relevant similarities). Now we can compute φ as:

$$\varphi = minimum_{\{A,B \in \mathcal{R}\}} Avg_{\{a \in A, b \in B\}} sim(a, b) \tag{7}$$

We use φ in Equation 6 to compute for all pairs of collections A and B the collection-related similarity, $sim_{col}(A, B)$. From this we can estimate the relevant similarity between the objects using Equation 3.

3.3 Applicability

Both the omission and the estimation method can be applied to all similarity-based clustering algorithms. Examples of such algorithms are given in the Sect. 4.

The more objects belong to the same collection, the more similarities are omitted by the omission method. This reduces the amount of data, which may lead to less accurate clusters. The estimation method bases its estimation of the collection-related similarity of a pair of collections on the average similarity between the objects in the collections. If the collections contain very few objects, the estimations become uncertain. Therefore, we expect that omission is the best choice for data sets with a large number of small collections and estimation is best for data sets with a small number of large collections. In Sect. 4 we test the effects of the number of collections in experiments.

The computational complexity of both methods is small, so that they scale very well to large clustering tasks. The extra complexity that the omission method adds to a clustering method is negligible. The only action it introduces is checking whether two objects are from the same collection. The estimation method does not change a clustering algorithm, but requires computing the relevant similarities. This can be done in two passes through the similarity matrix, so that the time complexity is $O((n-1)^2)$, where n is the number of objects.

4 Evaluation

We applied the omission and the estimation method to three commonly used clustering algorithms. The first algorithm is K-means [7], a partitional algorithm. We used a similarity-based version of K-means, which uses the average similarity between all objects in a cluster instead of the distance between objects and the cluster centre. K-means was run with 100 random initializations. The second algorithm was an agglomerative clustering algorithm: average linkage clustering [6]. This algorithm starts with each object in its own cluster and merges the two clusters with the smallest average similarity until the desired number of clusters is reached. The third algorithm was bisecting K-means [1], a divisive algorithm that starts with all objects in one cluster. In each step the cluster

Table 1. Properties of the data sets: the type of objects, the type of sets that form the collections, the type of clusters we want to find, the number of objects, the number of clusters in the gold standard and the number of collections

Data set	Object type	Col. type	Target cluster type	#Objects	#Clus	#Cols
hotel	web page	web site	topic	52	7	5
conference	web page	web site	topic	56	13	5
surf	web page	web site	topic	84	12	5
school	web page	web site	topic	105	17	5
no-flash	picture	location	item in picture	108	18	3
flash	picture	location	item in picture	108	18	6
artificial	-	-	-	400	40	1 to 10

with the largest number of objects is split into two. To split the clusters we used the similarity-based version of K-means with 20 random initializations.

The algorithms were evaluated on two types of real world data: web pages and pictures. The data sets can be downloaded from http://homepages.cwi.nl/∼vera/-clustering_data. In addition, we tested the influence of various characteristics of the data using artificial data. Table 1 shows the main properties of the data sets.

For the web site data sets the task of the clustering algorithms was to find clusters of web pages about the same topic in a number of comparable web sites. We used 4 data sets with sites from different domains: windsurf clubs, schools, small hotels and computer science conferences. Each data set consisted of the pages from 5 sites (collections). We manually constructed for each domain a gold standard: a set of page clusters that corresponded to topics (e.g. *pages listing important dates*). The gold standard clusters were not evenly spread over the sites: for many topics some of the sites contained multiple pages or zero pages. We used a standard similarity measure: the cosine of word frequency vectors [8].

The image data sets consisted of pictures of 18 small items, such as toys and fruit, taken at three different backgrounds (locations). The task was to find clusters of pictures of the same item. The 'no-flash' data set contained for each item two pictures per location taken without flash. All pictures taken at one location formed one collection. The 'flash' data set contained for each item and location one picture that was taken with flash and one that was taken without flash. Each combination of lighting and background formed a collection. We used a simple pixel-based similarity function that compared the RGB-values of the pixels that were at the same position in the pictures. The similarity between two pictures was computed as 1/total_RGB_difference. Of course, more advanced image-comparison techniques exist, but the goal of this paper is not to provide an optimal solution for clustering images.

For the artificial data sets we created 400 objects that were divided evenly over a number of collections (1 to 10). We created 40 gold standard clusters, in such a way that each cluster contained objects from all collections. The similarities between the objects were drawn from a normal distribution with a mean of 0.05 and a standard deviation of 0.075. When two objects were from the same cluster, we added 0.15 to the similarity, representing relevant similarity. For objects from

the same collection we added a certain amount of collection-related similarity. We called this amount ρ and experimented with various values for ρ. The mentioned values were chosen in such a way that they resembled the values that were found in the school data set. For each parameter setting 10 data sets were created. All reported numbers are averages over the 10 data sets.

4.1 Results

The standard and the enhanced clustering algorithms were applied to the data sets. The number of clusters was set equal to the actual number of clusters in the gold standards. For each cluster we counted the number of different collections from which the objects in the cluster originated. The standard clustering algorithms created clusters with very small numbers of collections: on average 1.0 to 2.3, where the gold standards had 4.1 to 6.0 collections per cluster. This confirms our hypothesis that standard algorithms tend to form clusters of objects from one collection. The enhanced methods frequently created clusters that spanned multiple collections. On average, the omission and the estimation method increased the number of collections per cluster by respectively 54% and 58%.

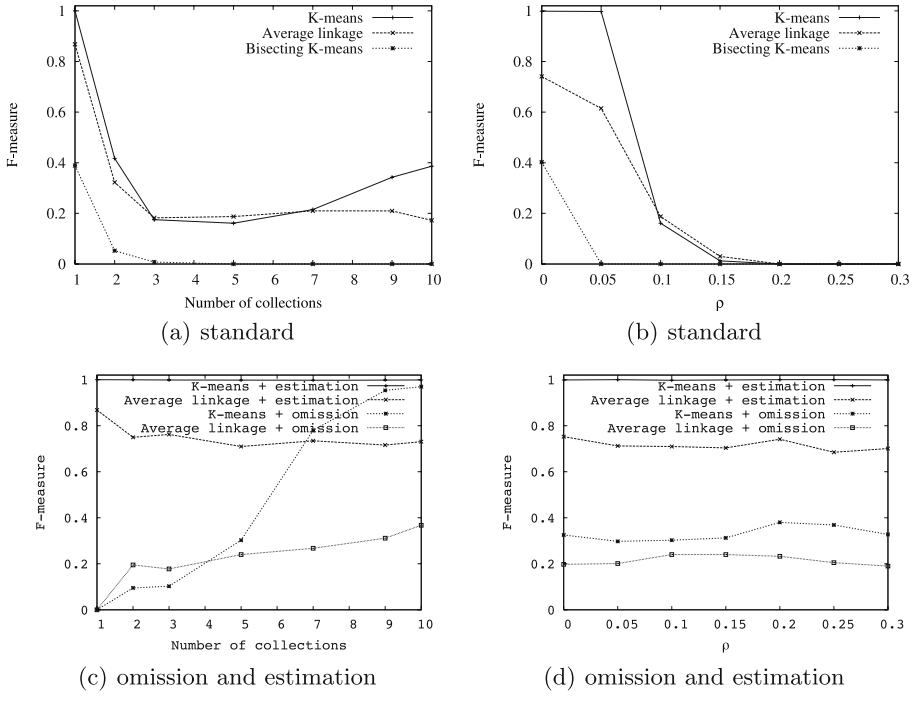
We evaluated the clusters through comparison with the gold standards. We counted how many of the created clusters also occurred in the gold standards and vice versa. A created cluster was considered to be the same as a cluster from the gold standard if more than 50% of the objects in the clusters were the same. The overlap of a set of clusters with a gold standard is expressed by F-measure [8].

Table 2 shows the F-measure of the standard and the enhanced clustering algorithms. On average, the omission method increased the F-measure by 93%. The estimation improved the average with 87%. The omission method gave excellent results with K-means, but performed less well with average linkage and bisecting K-means. A possible explanation for this is that divisive and agglomerative clustering algorithms are sensitive to incorrect choices that are made in the early steps of the clustering process. Omitting similarities may remove information that is essential at this stage. The estimation method led to fairly large improvements with all three clustering methods.

To test the effects of the number of collections, we generated data sets with 1 to 10 collections. ρ was 0.1. Fig. 1a shows the F-measure of the standard

Table 2. F-measure of the clusters created by the standard algorithms (std.), with the omission method (omi.) and with the estimation method (est.). Best scores are bold.

Data set	K-means			Average linkage			Bisecting K-means		
	std.	omi.	est.	std.	omi.	est.	std.	omi.	est.
hotel	0.14	**0.43**	0.29	0.00	**0.29**	0.14	0.14	**0.18**	0.14
conference	0.15	**0.61**	0.31	0.00	0.08	**0.15**	0.00	**0.15**	**0.15**
surf	0.00	**0.42**	0.00	0.00	0.00	0.00	**0.25**	0.12	**0.25**
school	0.20	**0.41**	0.38	0.12	0.08	**0.18**	0.06	**0.15**	0.12
no-flash	0.25	**0.47**	0.33	0.11	0.00	**0.13**	0.22	0.00	**0.50**
flash	0.11	0.11	**0.13**	0.06	0.06	**0.11**	0.06	0.07	**0.17**



(a) standard (b) standard

(c) omission and estimation (d) omission and estimation

Fig. 1. F-measure of the clusters created by the standard algorithms, with the omission method and with the estimation method on artificial data sets with various numbers of collections and various amounts of collection-related similarity (ρ).

algorithms. When the objects came from several collections, the algorithms could not distinguish between collection-related similarity and relevant similarity and often produced clusters that coincided with collections. The results of the enhanced methods are shown in Fig. 1c (results of bisecting K-means are similar). In Sect. 3 we postulated the hypothesis that omission works best if we have a large number of collections. Fig. 1c shows that this is indeed the case. As expected, the performance of the estimation method deteriorated with increasing numbers of collections. However, the effect is very small. Apparently, with 10 collections, there was still enough data to accurately estimate collection-related similarity.

Next, we tested the influence of the strength of the collection-related similarity by varying the value of parameter ρ. The data sets in these experiments had 5 collections. From Fig. 1b we can see that for the standard algorithms more collection-related similarity led to lower F-measures. Fig. 1d shows that with the omission and the estimation method the performance of the algorithms was stable even when there was a large amount of collection-related similarity.

In sum, these experiments show that commonly used similarity measures do not lead to satisfying results when the data comes from multiple collections. In this situation using collection information improves clustering.

5 Conclusions

In many clustering tasks the data come from more than one collection. In this paper we showed that if we want to find clusters that span multiple collections, standard similarity measures and clustering methods are not adequate.

We provided two methods to suppress similarity associated with collections: omission, a modification to clustering algorithms and estimation, a modification to similarity measures. The methods do not require any domain-specific knowledge and can be applied on top of all clustering algorithms that use similarities between objects. Our experiments show that clustering methods enhanced with one of these methods can effectively find clusters in data from multiple collections. Compared to the standard clustering algorithms, the enhanced methods created clusters that spanned more collections and were more similar to gold standard clusters created by humans. On average, the omission and the estimation method increased the number of collections per cluster by respectively 54% and 58%. The average overlap with the gold standards was increased by 93% and 87%. Both the omission and the estimation method proved effective, but on artificial data and most of the real world data sets estimation outperformed omission. This shows that the more fined-grained analysis of the estimation method is effective despite the additional assumptions that underlie this analysis.

Future work includes generalizing our approach to situations where there is more than one type of collection. In these cases, we have multiple collection labels that all correspond to collection-related similarity. Most likely, improvements can be made by compensating the collection-related similarity of each collection type separately, instead of treating each combination of collections as one collection.

References

1. Steinbach, M., Karypis, G., Kumar, V.: A comparison of document clustering techniques. In: KDD Workshop on Text Mining, Boston, MA (2000)
2. Sahoo, N., Callan, J., Krishnan, R., Duncan, G., Padman, R.: Incremental hierarchical clustering of text documents. In: CIKM 2006, Arlington, VA, pp. 357–366 (2006)
3. Li, X., Liu, B.: Learning from positive and unlabeled examples with different data distributions. In: Gama, J., Camacho, R., Brazdil, P.B., Jorge, A.M., Torgo, L. (eds.) ECML 2005. LNCS (LNAI), vol. 3720, pp. 218–229. Springer, Heidelberg (2005)
4. Huang, Y., Mitchell, T.M.: Text clustering with extended user feedback. In: SIGIR 2006, Seattle, WA, USA, pp. 413–420 (2006)
5. Bronstein, A.M., Bronstein, M.M., Bruckstein, A.M., Kimmel, R.: Partial similarity of objects, or how to compare a centaur to a horse. International Journal of Computer Vision (in press)
6. Voorhees, E.M.: Implementing agglomerative hierarchical clustering algorithms for use in document retrieval. Information Processing and Management 22(6), 265–276 (1986)
7. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. I, pp. 281–297 (1967)
8. Salton, G., McGill, M.J.: Introduction to modern information retrieval. McGraw-Hill, New York (1983)