Using *k*-Nearest Neighbor and Feature Selection as an Improvement to Hierarchical Clustering

Phivos Mylonas, Manolis Wallace, and Stefanos Kollias

School of Electrical and Computer Engineering
National Technical University of Athens
9, Iroon Polytechniou Str., 157 73 Zographou, Athens, Greece
{fmylonas,wallace}@image.ntua.gr
stefanos@cs.ntua.gr

Abstract. Clustering of data is a difficult problem that is related to various fields and applications. Challenge is greater, as input space dimensions become larger and feature scales are different from each other. Hierarchical clustering methods are more flexible than their partitioning counterparts, as they do not need the number of clusters as input. Still, plain hierarchical clustering does not provide a satisfactory framework for extracting meaningful results in such cases. Major drawbacks have to be tackled, such as curse of dimensionality and initial error propagation, as well as complexity and data set size issues. In this paper we propose an unsupervised extension to hierarchical clustering in the means of feature selection, in order to overcome the first drawback, thus increasing the robustness of the whole algorithm. The results of the application of this clustering to a portion of dataset in question are then refined and extended to the whole dataset through a classification step, using k-nearest neighbor classification technique, in order to tackle the latter two problems. The performance of the proposed methodology is demonstrated through the application to a variety of well known publicly available data sets.

1 Introduction

The essence of clustering data is to identify homogeneous groups of objects based on the values of their attributes. It is a problem that is related to various scientific and applied fields and has been used in science and in the field of data mining for a long time, with applications of techniques ranging from artificial intelligence and pattern recognition to databases and statistics [1]. There are different types of clustering algorithms for different types of applications and a common distinction is between hierarchical and partitioning clustering algorithms. But although numerous related texts exist in the literature, clustering of data is still considered an open issue, basically because it is difficult to handle in the cases that the data is characterized by numerous measurable features. This is often referred to as the curse of dimensionality.

Although hierarchical clustering methods are more flexible than their partitioning counterparts, in that they do not need the number of clusters as an input, they are less

G.A. Vouros and T. Panayiotopoulos (Eds.): SETN 2004, LNAI 3025, pp. 191–200, 2004.

robust in some other ways. More specifically, errors from the initial steps of the algorithm tend to propagate throughout the whole procedure to the final output. This could be a major problem, with respect to the corresponding data sets, resulting to misleading and inappropriate conclusions. Moreover, the considerably higher computational complexity that hierarchical algorithms typically have makes them inapplicable in most real life situations, due to the large size of the data sets.

Works in the field of classification focus in the usage of characterized data, also known as training data, for the automatic generation of systems that are able to classify (characterize) future data. This classification relies on the similarity of incoming data to the training data. The main aim is to automatically generate systems that are able to correctly classify incoming data [1].

Although the tasks of classification and clustering are closely related, an important difference exists among them. While in the task of classification the most important part is the distinction between classes, i.e. the detection of class boundaries, in the task of clustering the most important part is the identification of cluster characteristics. The latter is usually tackled via the selection of cluster representatives or cluster centroids).

Typically, in order to achieve automatic classification systems generation, one first needs to detect the patterns that underlie in the data, in contrast to simply partitioning data samples based on available labels [7], and then study the way these patterns relate to meaningful classes. Efficient solutions have been proposed in the literature for both tasks, for the case in which a unique similarity or dissimilarity measure is defined among input data elements [6]. When, on the other hand, multiple independent features characterize data, and thus more than one meaningful similarity or dissimilarity measures can be defined, both tasks become more difficult to handle. A common approach to the problem is the lowering of input dimensions, which may be accomplished by ignoring some of the available features (feature selection) [2].

In the case when input features are independent, or when the relation among them is not known a priori, which is often the case with real data, a decrease of space dimensions cannot be accomplished without loss of information. The proposed algorithm of this work is an extension of agglomerative clustering in this direction and is based on a soft selection of features to consider when comparing data. The results of the initial clustering, performed on a small amount of the original data set, are then refined via a classification step; this step, although unsupervised, is based on the principles of the k-nearest neighbour classification scheme and is applied to the whole data set. In this way we overcome two major drawbacks that dominate agglomerative clustering; the one of initial error propagation and the one regarding complexity issues. This important step also contributes to the experimental evaluation of the method's efficiency.

The structure of the paper is as follows: in section 2, after a short introduction to agglomerative clustering, we present the main problems that are related to our task and the proposed method for initial clustering. In section 3 we explain how a k-nearest neighbour classifier can be used to refine, as well as to experimentally verify the efficiency of the algorithm. Finally, in section 4, we present experimental results for the proposed algorithm and in section 5, we present our concluding remarks.

2 Agglomerative Clustering and Soft Feature Selection

Most clustering methods belong to either of two general methods, partitioning and hierarchical. Partitioning methods create a crisp or fuzzy clustering of a given data set, but require the number of clusters as input. When the count of patterns that exist in a data set is not known beforehand, partitioning methods are inapplicable; an hierarchical clustering algorithm needs to be applied.

Hierarchical methods are divided into agglomerative and divisive. Of those, the first are the most widely studied and applied, as well as the most robust. Their general structure is as follows [4]:

1. Turn each input element into a singleton, i.e. into a cluster of a single element.
2. For each pair of clusters c_1, c_2 calculate their distance $d(c_1, c_2)$.
3. Merge the pair of clusters that have the smallest distance.
4. Continue at step 2, until the termination criterion is satisfied. The termination criterion most commonly used is the definition of a threshold for the value of the distance.

The two key points that differentiate agglomerative methods from one another, and determine their efficiency, are the distance and the termination criterion used. Major drawbacks of agglomerative methods are their high complexity and their susceptibility to errors in the initial steps, that propagate all the way to their final output.

The core of the above generic algorithm is the ability to define a unique distance among any pair of clusters. Therefore, when the input space has more than one dimensions, an aggregating distance function, such as Euclidean distance, is typically used [9]. This, of course, is not always meaningful and there are cases where a selection of meaningful features needs to be performed, prior to calculating a distance [8]. In other words, it may not be possible to select a single distance metric, which will apply in all cases, for a given data set. Moreover, one feature might be more important than others, while all of the features are useful, each one to its own degree.

In this paper we tackle feature weighting based on the following principle: while we expect elements of a given meaningful set to have random distances from one another according to most features, we expect them to have small distances according to the features that relate them. We rely on this difference in distribution of distance values in order to identify the context of a set of elements, i.e. the subspace in which the set is best defined.

More formally, let c_1 and c_2 be two clusters of elements. Let also r_i, $i \in \mathbb{N}_F$ be the metric that compares the *i*-th feature, and F the overall count of features (the dimension of the input space). A distance measure between the two clusters, when considering just the *i*-th feature, is given by:

$$f_i(c_1, c_2) = \sqrt[\kappa]{\frac{\sum_{a \in c_1, b \in c_2} \left[r_i(a_i, b_i) \right]^\kappa}{|c_1||c_2|}} \tag{1}$$

where the subscript *i* denotes the *i*-th feature of an element, |c| is the cardinality of cluster c and $\kappa \in \mathbb{R}$ is a constant. Typical value used for κ is 2. The overall distance between c_1 and c_2 is calculated as:

$$d(c_1, c_2) = \sum_{i=1}^{F} \left[x_i(c_1, c_2) \right]^{\lambda} f_i(c_1, c_2) \tag{2}$$

where x_i is the degree to which i, and therefore f_i, is included in the soft selection of features, $i \in N_F$ and $\lambda \in \mathbb{R}$ is a constant. Typical value used for λ is 2. Based on the principle presented above, values of vector x are selected through the minimization of distance *d* [12]. The features that relate c_1 and c_2 are "most probably" the ones that produce the smallest distances f_i.

3 Refinement and Classification through *k*-Nearest Neighbor Classification

As stated in preceding sections, the primary aim of clustering algorithms is not to correctly classify data, but rather to identify the patterns that underlie in it and produce clusters of similar data samples. Therefore, 'wrong' elements in clusters may be acceptable, as long as the overall cluster correctly describes an existing and meaningful pattern: in fact, we have established in our previous work that clusters with wrongfully assigned data samples may be better than perfect data set partitionings in describing the underlying patterns and thus may lead to better classifier initialization [7]. This implies that if we feed labelled data to the algorithm and measure the classification rate may not be enough to evaluate the actual efficiency of the algorithm.

In order for a clustering algorithm to be properly evaluated, the patterns described by the clusters in its output need to be evaluated; their application towards the generation of a classifier and the evaluation of the resulting classifier is a means towards this direction. In this paper we examine whether the specific results of such an algorithm, applied to several well known machine learning data sets, are meaningful by evaluating the results from a k-nearest neighbours classifier that is created by using them.

Undoubtfully, several classification schemes exist in the literature [3]. We have chosen to work with the k-nearest neighbours (kNN) classifier, although others could have been chosen as well, mainly because of the nature of the instance-based learning method itself and its straightforward approach [11]. The kNN algorithm is extremely simple, yet powerful, used in many applications and can be safely applied in all sorts of data sets, real life and artificial ones, independently of size or time compromises, resulting into high quality scientific observations. kNN is also extremely suitable to use in cases where instances map to points in $\mathbb{R}^n$, there are lots of training data into consideration and – after performing soft feature selection – less than 20 attributes per instance.

Possible disadvantages to the kNN method, acknowledging the fact that it typically considers *all* the attributes from *all* the elements, are easily overcome by applying the

initial clustering procedure on a small subset of the available data, thus reducing the number of elements that the classification scheme will need to consider in order to classify each incoming sample. The aforementioned approach is extremely suitable and appropriate for online classification.

Specifically, the kNN algorithm assumes that all elements correspond to points in the n-dimensional space R^n. The neighbours of an element are defined in form of some distance measurement. A variety of metrics can be used as distances in the algorithm, like Euclidean square distance:

$$d(a,b) = \sqrt{\sum_{i \in N_F} (a_i - b_i)^2} \tag{3}$$

Minkowsky distance:

$$d(a,b) = \sum_{i \in N_F} |a_i - b_i| \tag{4}$$

minimax distance:

$$r(a,b) = \max \sum_{i \in N_F} |a_i - b_i| \tag{5}$$

Mahalanobis distance and others.

Specifically, we tackle each initial element and calculate its distance from every other element in the data set. We define a priori the number of the nearest to the element under consideration neighbours, k, that are going to play a significant role in the cluster characterization of the element at the latest stage, thus using a suitable threshold regarding the precision of the classification. Clearly if k becomes very large, then the classifications will become all the same. Generally, there is some sense in making $k > 1$, but certainly little sense in making k equal to the number of training elements.

Formally, let e_q be each given query element to be classified and e_1, e_2 ,…e_k denote the k elements that are nearest to e_q. Let also c(a) be defined as:

$$c(a, j) = \begin{cases} 1 & , \text{a belongs to class j} \\ 0 & , otherwise \end{cases}$$

Then, e_q is classified as follows:

$$c(e_q) = z : \sum (e_i, z) = \max(\sum_{j=1}^{countofclasses} c(e_i, j)) \tag{6}$$

where e_i is the training instance (element) nearest to e_q. In other words, e_q is classified to the class to which most of its k closest neighbors belong.

Obviously, in order to apply the kNN classification scheme, a small set of labelled data samples are needed. In this work, we describe the unsupervised classification of data, and thus we assume such information to be unavailable; we only use data labels in our experiments in order to measure the classification rate and thus the performance of the algorithm. Therefore, we assume that each one of the clusters detected during the step of hierarchical clustering corresponds to a distinct class.

Using the classification scheme described above, and the cluster assignments of the clustered data samples as class labels, we may proceed to classify all available data elements. If the initial clustering was successful in revealing the patterns that underlie in the data, then this process will refine the output and improve the classification rate by removing some of the clusters' members that were a result of errors in the initial steps. Thus, this process offers an indication of the hierarchical clustering's true performance. Moreover, it makes the overall algorithm more robust, as opposed to simple hierarchical clustering, as it is more resilient to errors in the initial steps.

Finally, it is this step of classification that extends the findings of the initial clustering to the whole data set, thus allowing for the former to be applied on just a portion of the data set. This is very important, as without this it would not be possible to have the benefits of hierarchical clustering when dealing with larger data sets. Furthermore, a significant role in the classification process plays the iterative nature of the algorithm, which rises from the fact that the input is the same as the output, thus allowing several iterative applications of the algorithm, until the cluster assignments of the elements remain unchanged.

4 Experimental Results

In this section we list some indicative experimental results of the proposed methodology from application to real data sets from the well-known machine learning databases. In all consequent experiments we have used the Euclidean distance for the estimation of the k nearest neighbours. Values of κ, λ and k differ from case to case and are thus mentioned together with each reported result.

In all experiments the proposed clustering algorithm that is described in section 2 has been applied on a small portion of the data set, while the whole data was consequently classified based on the output of this step and applying kNN classification, as described in section 3.

Iris Data

The iris data set contains 150 elements, characterized by 4 features, that belong to three classes; two of these classes are not linearly separable from each other. The labels of the elements were not used during clustering and classification; there were used, though, for the estimation of the classification rates; specifically, each cluster was assigned to the class that dominated it. Results are shown in Tables 1 and 2, whereas the numbers inside parenthesis separated by commas denote the elements belonging to its one of the three classes in each step.

For the application of the proposed methodology a portion of the dataset, specifically 20% of it, was separated and submitted to the clustering procedure. The classification rate on this portion of the dataset (63.3%) is not impressive. Still, the application of the classification step on the whole data set produces a considerably better classification rate, which indicates that the initial clustering process had successfully

detected the patterns and the kNN classification process successfully clustered the remaining data.

We can also observe that the proposed methodology, although applying the computationally expensive step of hierarchical clustering to only 20% of the dataset (initial clustering for 30 elements), does not produce inferior results to the approach that applies an hierarchical clustering algorithm to the whole dataset. Comparing them to simple agglomerative clustering with no feature selection and no recursive classification (i.e. classification rate ~ 74%), proves its very good overall performance.

Table 1. Classification rates for Iris data (constants: $\kappa = \lambda = 1.3$, neighbours k = 5).

Method	cluster 1	cluster 2	cluster 3	Classification rate
Initial clustering	(2,0,4)	(6,1,4)	(2,9,2)	63,3%
Knn classification	(7,0,31)	(43,0,19)	(0,50,0)	82,7%

Table 2. Comparison of proposed classification approach to plain clustering.

Method	Classification rate
Clustering of the whole dataset	74,7%
Proposed approach	82,7%

Wisconsin Breast Cancer Database

The Wisconsin breast cancer database contains 699 elements, which are characterized by the following attributes: clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, mitoses. All these attributes assume integer values in [9]. Elements are also accompanied by an id, and class information; possible classes are benign and malignant. 65.5% of the elements belong to the benign class and 34.5% to the malignant class. 16 elements are incomplete (an attribute is missing) and have been excluded from the database for the application of our algorithm.

Detailed results acquired using the proposed methodology are available in Tables 3 and 4, whereas the numbers inside parenthesis separated by comma denote the elements belonging to its one of the two classes in each step. It is worth noting that, similarly to the case of iris data, although the classification rate of the initial clustering procedure, which was performed on a 7,32% subset of the original data set (50 data samples), is not extremely high, the classification step on the whole database refines it considerably. This indicates that the proposed clustering approach was efficient in revealing the patterns in the small portion of the data set, and the kNN process successfully utilized this information for the refinement of the clustering and the extension to the remaining dataset.

Additionally, performing the initial clustering on a mere 7,32% subset is not only more efficient computationally wise, it is also better in the means of quality and per-

formance, as indicated by the results in Table 4, when compared to the approach of applying the hierarchical process to the whole data set.

Finally, it is worth noting that the small computational needs of the kNN classification process allow for its repeated / recursive application on the data. Such reclassification steps also induce an increase to the classification rate, as is evident in Table 3, thus further stressing the efficiency of the proposed approach in revealing the patterns that underlie in the data. The classification rate of 93.1% that is reported is extremely high for this data set for an unsupervised clustering algorithm.

Table 3. Classification rates for Wisconsin data (constants: $\kappa = 1.3$, $\lambda = 1.3$, $k = 3$).

Method	cluster 1	cluster 2	cluster 3	Classification rate
Initial clustering	(21,2)	(9,10)	(0,8)	78,0%
Knn classification	(341,24)	(100,50)	(3,165)	88,7%
Knn reclassification 1	(348,21)	(91,31)	(5,187)	91,7%
Knn reclassification 2	(349,20)	(90,23)	(5,196)	93,0%
Knn reclassification 3	(349,20)	(90,22)	(5,197)	93,1%

Table 4. Comparison of proposed classification approach to plain clustering.

Method	Classification rate
Clustering of the whole dataset	86,1%
Proposed approach	88,7%
Proposed approach with recursive kNN	93,1%

This performance is not far from that of trained classification systems that utilize the same dataset. This is indicative of the method's efficiency, considering that we are referring to the comparison of an unsupervised method to a supervised ones. Best results may be presented in our work in [12], but there was undoubtfully more information used, mainly because a Gaussian distribution of the dataset was assumed, which is not the case in this work. Furthermore, we must also note that number k of the nearest neighbours is obviously chosen based on observed relative statistics and is subject to further improvements.

5 Conclusions

In this paper we developed an algorithm for the detection of patterns in unlabelled data in the means of agglomerative clustering improvement, using the k-nearest neighbours classification scheme. The first step of the algorithm consists of an hierarchical clustering process, applied only to a subset of the original data set. This process

performs a soft feature selection in order to determine the subspace within which a set of elements is best defined and thus it is suitable for data sets that are characterized by high dimensionality. The second part of the algorithm performs a k-nearest neighbours classification. This process considers initial clusters to be labels and uses this information to build a classifier, through which to classify all data. Thus, errors from the hierarchical algorithm's initial steps are corrected; moreover, as the computational complexity of this classification step is considerably smaller that that of the complexity of the clustering process, it may be applied to the entire dataset. In addition to making the overall algorithm more efficient and resilient to errors, it also serves as a means for its evaluation.

The efficiency of the proposed algorithm has been demonstrated through application to a variety of real data sets. Experiments on the iris dataset indicated the method's ability to perform as well as simple hierarchical clustering having a much better complexity. Application on the Wisconsin breast cancer database which is a multi – dimensional data set, on the other hand, was indicative of the method's performance in such environments: the results of the application of the proposed methodology to less than 10% of the available data exceed those obtained by application of the computationally expensive hierarchical clustering process to the entire dataset.

In our future work we aim to extend on our work on improvement of the hierarchical clustering process by providing guidelines for the automatic selection of the thresholds used in this work, namely parameters κ and λ of the clustering process and k of the kNN classification. On a more practical side, we are already working towards the application of the methodology presented herein for the clustering of usage history and the extraction of low level and semantic user preferences, in the framework of the EU funded IST-1999-20502 FAETHON project.

Acknowledgments

This work has been partially funded by the EU IST-1999-20502 FAETHON project.

References

1. Hirota, K., Pedrycz, W. (1999) Fuzzy computing for data mining. Proceedings of the IEEE 87:1575–1600.
2. Kohavi, R., Sommerfield, D. (1995) Feature Subset Selection Using theWrapper Model: Overfitting and Dynamic Search Space Topology. Proceedings of KDD-95.
3. Lim, T.-S., Loh, W.-Y., Shih, Y.-S. (2000) A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-three Old and New Classification Algorithms. Machine Learning 40:203–229.
4. Miyamoto, S. (1990) Fuzzy Sets in Information Retrieval and Cluster Analysis. Kluwer Academic Publishers.
5. Swiniarski, R.W., Skowron, A. (2003) Rough set methods in feature selection and recognition. Pattern Recognition Letters 24:833–849.

6. Theodoridis, S. and Koutroumbas, K. (1998) Pattern Recognition, Academic Press.
7. Tsapatsoulis, N., Wallace, M. and Kasderidis, S. (2003) Improving the Performance of Resource Allocation Networks through Hierarchical Clustering of High – Dimensional Data. Proceedings of the International Conference on Artificial Neural Networks (ICANN), Istanbul, Turkey.
8. Wallace, M., Stamou, G. (2002) Towards a Context Aware Mining of User Interests for Consumption of Multimedia Documents. Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), Lausanne, Switzerland.
9. Yager, R.R. (2000) Intelligent control of the hierarchical agglomerative clustering process. IEEE Transactions on Systems, Man and Cybernetics, Part B 30(6): 835–845 Tsapatsoulis, N., Wallace, M. and Kasderidis, S.
10. Wallace, M., Mylonas, P. (2003) Detecting and Verifying Dissimilar Patterns in Unlabelled Data. 8th Online World Conference on Soft Computing in Industrial Applications, September 29th - October 17th, 2003.
11. Tom M. Mitchell. Machine Learning. McGraw-Hill Companies, Inc., 1997.
12. Wallace, M. and Kollias, S., "Soft Attribute Selection for Hierarchical Clustering in High Dimensions", Proceedings of the International Fuzzy Systems Association World Congress(IFSA), Istanbul, Turkey, June-July 2003.