# Soft Attribute Selection for Hierarchical Clustering in High Dimensions *

**Manolis Wallace**
Department of Computer Science
National Technical University of Athens
wallace@image.ntua.gr

**Stefanos Kollias**
Department of Computer Science
National Technical University of Athens
stefanos@cs.ntua.gr

## Abstract

In this paper we perform an hierarchical clustering in high – dimensional spaces, without first applying any space reduction. Instead, in each step of the algorithm we perform a soft feature selection, witch does not have to be shared among all input elements. The main goal is to correctly identify the patterns that underly in the data. The proposed algorithm is applied, with promising results, in a well known and widely studied set of medical data.

**Keywords:** Hierarchical clustering, high dimensions, feature selection.

## 1 Introduction

Automatic analysis of data for extraction of information, and eventually knowledge, has been a quest for long. Researchers in the field of *knowledge extraction* have worked in this direction for more than a decade, producing important results; still, this remains an open issue [1].

Efficient solutions have been proposed in the literature for this task, for the case in which a unique similarity or dissimilarity measure is defined among input data elements [2]. When, on the other hand, multiple independent features characterize data, and thus more than one meaningful similarity or dissimilarity measures can be defined, the task becomes more difficult to handle. Especially for the case when the count of features is great, the meaningful analysis of data is almost impossible. This is known as the *dimensionality curse*.

A common approach to the dimensionality curse is the lowering of input dimensions [3]. This may be accomplished by ignoring some of the available features, or by applying some space transformation. In the case when input features are not independent from each other, a decrease of dimensions is very helpful. On the other hand, when input features are independent, or when the relation among them is not known a priori, a decrease of space dimensions cannot be accomplished without loss of information.

Therefore, if the relation among features is not known before hand, and the aim is to detect the patterns that exist in the data, the decrease of dimensions is not possible. In this work we attempt to tackle such a problem: we focus on the detection of patterns in high – dimensional data, when the count of distinct patterns in the data and the relation among input features are unknown. Our approach is based on a soft selection of features to consider when comparing data. This selection is dependant on the data in question. The proposed algorithm is an extension of agglomerative clustering.

## 2 Soft Feature Selection and Clustering in High – Dimensional Spaces

The source of the dimensionality curse is that elements are usually grouped together based on their similarity in a single or a few features. When the

*This work has been partially funded by the ORESTEIA IST-2000-26091/TBD project

total number of features is high, small distances in a small subset of them barely affect the overall distance, when an aggregation of distances in all features is used. Thus, only when the correct subset of features is considered, can elements be compared correctly.

In this paper, we tackle feature selection based on the following principle: while we expect elements of a given 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 *context*, i.e. the features that most probably relate a set of elements.

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

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

where $e_i$ is the $i$-th feature of element $e$, $|c|$ is the cardinality of cluster $c$ and $\kappa \in R$ is a constant.

The context is a selection of features to consider when calculating an overall distance value. We can define it as a fuzzy set $x$ defined on $N_F$, with a scalar cardinality of one. Then the overall distance between $c_1$ and $c_2$ is calculated as

$$d(c_1, c_2) = \sum_i [x_i(c_1, c_2)]^\lambda f_i(c_1, c_2) \qquad (2)$$

where $i \in N_F$, $\lambda \in R$ is a constant and $x_i$ is the degree to which $i$, and therefore $f_i$, is included in the context.

According to the principle presented in the beginning of this paragraph, the features that relate $c_1$ and $c_2$ are probably the ones that produce the smallest distances $f_i$. Therefore, the "correct" context can be calculated as the context that produces the best (smallest) overall distance.

When $\lambda = 1$ the solution is trivial: the feature that produces the smallest distance is the only one selected. The degree to which it is selected is 1. If more than one features produce the best distance, then they are equally selected, as there is no information as to which should be favored.

When $\lambda \neq 1$ and $f_i(c_1, c_2) \neq 0 \forall i \in N_F$, then it is easy to prove that the best context is given by:

$$x_1(c_1, c_2) = \frac{1}{\sum_i [\frac{f_1(c_1,c_2)}{f_i(c_1,c_2)}]^{\frac{1}{\lambda-1}}} \qquad (3)$$

$$x_i(c_1, c_2) = x_1 [\frac{f_i(c_1, c_2)}{f_1(c_1, c_2)}]^{\frac{1}{\lambda-1}}$$

where $i \in N_F$. Proof is omitted for the sake of space.

When $\lambda \neq 1$ and $\exists i \in N_F : f_i(c_1, c_2) = 0$, then the features for which $f_i(c_1, c_2) = 0$ are the ones the are (equally) selected.

As $\lambda$ increases, pairs of clusters that are related by fewer features, and thus have greater values in their contexts, are obviously assigned smaller distances. In order for distances to be used for cluster comparison, in the process of agglomerative clustering, it is imperative that they are transformed as to become directly comparable to each other, even when different contexts are used for different pairs of clusters. Therefore, the following metric is used:

$$CI(c_1, c_2) = \frac{d(c_1, c_2)}{x_\lambda(c_1, c_2)}$$

$$x_\lambda(c_1, c_2) = \sum_i [x_i(c_1, c_2)]^\lambda$$

We often refer to this metric as a compatibility indicator among clusters. When features are quantized to a small set of levels, as is often the case with digital data, cases for which $f_i(c_1, c_2) = 0$ are not rare. Especially in the first steps of agglomerative clustering, when clusters are of small size, the best $CI$ is almost always zero. Since errors in the initial steps of agglomerative clustering propagate all the way to the final output, it is important to always make the best selection possible for the pair of clusters to merge. Therefore, especially for the case of $CI$s that are equal to zero, we introduce one more criterion: out of all the pairs that have $CI = 0$, we will always select the

one that has zero distances for the most features. In other words, out of all the pairs of similar clusters, we select the ones that are similar according to the greatest number of features.

For the process of agglomerative clustering to be fully defined, in addition to the aforementioned metric of cluster distances, a termination criterion is needed [2]. In this work, a threshold on the value of $CI$ can be used. This is meaningful, as the $CI$ is increasing as we move from one step to the next. Proof is again omitted for the sake of space.

This way, the algorithm gradually groups elements together, based on their similarities; for each cluster, a different subset of features may be considered for the calculation of similarities. The average values of features for each cluster form the centroid, i.e. a "virtual" element that is located in the center of the cluster, when all of its elements are placed in the $F$-dimensional space. Its position may be considered as a description of the feature values of the pattern that this cluster corresponds to.

Of course, not all features are equally important when describing a pattern. The same principle as in the calculation of context in equation 3 can be used for the soft selection of the set of features that matter the most for each pattern: if $g_i(c) = \sum_{a,b \in c}[f_i(a,b)]^\kappa$ , $i \in N_F$, then the cluster's context $xc$ can be defined as the context that minimizes the following:

$$G(c) = \sum_i [xc_i)]^\lambda g_i(c) \tag{4}$$

In other words, the cluster's context is the context with respect to which the cluster's elements are most similar to one another.

## 3 Experimental Results

The algorithm has been applied to the Wisconsin breast cancer database. This 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 $[1, 10]$. 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. In the past this database has been used extensively for the generation of systems for automatic detection of breast cancer.

Table 1: The clusters that the algorithm detected, for different values of $\kappa$, $\lambda$.

| $\kappa = \lambda = 2$ | cluster 1 | cluster 2 | cluster 3 |
|---|---|---|---|
| benign | 218 | 192 | 34 |
| malignant | 5 | 56 | 178 |

| $\kappa = \lambda = 5$ | cluster 1 | cluster 2 | cluster 3 |
|---|---|---|---|
| benign | 192 | 3 | 249 |
| malignant | 56 | 154 | 29 |

In this work, we use the whole set as input of the algorithm. The class labels are not included in the input. The aim is to test whether the automatically detected patterns are consistent with the known patterns that exist in the data. In table 1 we present the three clusters generated by the algorithm for different values of $\kappa$ and $\lambda$. We can observe that each one of the clusters corresponds to a great extend to either malignant or benign elements. In other words, the detected patterns correspond to real patterns that are known to exist in the data, although the input data set is linearly inseparable[4].

In order to verify that the pattern is extracted correctly, we perform one more test. We compare all elements in the data set to the centroid of the malignant cluster. The comparison is performed using the distance metric of equation 2. The context used is the context of the cluster (it is calculated via the minimization of equation 4). This can be considered as querying the data set, using the centroid and the context as query parameters. Using a manually set threshold, we select the elements that have a small distance from the centroid. In table 2 we present precision and recall values for these queries. High values of both

Table 2: How the detected patterns relate to the known classification to benign and malignant elements.

| | threshold | recall | precision |
|---|---|---|---|
| $\kappa = \lambda = 2$ | 3 | 86.19% | 90.35% |
| $\kappa = \lambda = 5$ | 3.6 | 98.33% | 94.71% |

recall and precision indicate that, although benign elements exist in the detected clusters, the pattern is still extracted correctly.

Although class information was not used as input for the generation of the system that queries the data set, results are similar in recall and precision to those reported in the literature in other works [5], [6]; these works, in contrast to the one presented herein, use a part of the labelled data set as training data in order to create a classifier.

From all the above, it is apparent that the proposed algorithm is successful in revealing unknown patterns in data sets of high dimensions.

The source code of the programs used to test the proposed algorithm, together with a copy of the used data set, can be found at [7].

## 4 Conclusions and Future Work

In this paper we performed an hierarchical clustering in high – dimensional spaces. This was based on an on-line soft selection of features to consider when comparing clusters. The efficiency of the algorithm was demonstrated through its application on a medical data set. The proposed algorithm can be used to detect and extract unknown patterns in high – dimensional unlabelled input data.

The results of the presented algorithm are dependent on the selection of kappa, lambda and termination criterion thresholds. Currently, the selection of such thresholds is manual; it is part of our future work to explore ways to automatically select them. As part of our future work, we also intend to apply our algorithm on other data sets, in order to further verify its efficiency, and study whether the selection of thresholds is closely related to the data set in question.

## Acknowledgments

The breast cancer database was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg [8].

The authors would also like to express their gratitude to Prof. Nikhil Pal and Prof. Sikha Bagui for their help in acquiring the database.

## References

[1] Hirota, K., Pedrycz, W. (1999). Fuzzy computing for data mining. Proceedings of the IEEE, 87, 1575–1600.

[2] Theodoridis, S. and Koutroumbas, K. (1998). Pattern Recognition, Academic Press.

[3] Kohavi, R., Sommerfield, D. (1995). Feature Subset Selection Using the Wrapper Model: Overfitting and Dynamic Search Space Topology. Proceedings of KDD-95.

[4] Bennett, K.P., Mangasarian, O.L. (1992). Robust linear programming discrimination of two linearly inseparable sets. Optimization Methods and Software, 1, 23–34.

[5] Bagui, S.C., Bagui, S., Pal, K., Pal, N.R. (2003). Breast cancer detection using rank nearest neighbor classification rules. Pattern Recognition, 36, 25–34.

[6] Zhang, J. (1992). Selecting typical instances in instance-based learning. Proceedings of the Ninth International Machine Learning Conference Aberdeen, Scotland, 470–479.

[7] The implementation of the proposed algorithm in Java:
`http://www.image.ntua.gr/~wallace/cluster/code`

[8] Wolberg, W.H., Mangasarian, O.L. (1990). Multisurface method of pattern separation for medical diagnosis applied to breast cytology. Proceedings of the National Academy of Sciences, bf 87, 9193–9196.