A COMPARISON BETWEEN K-MEANS AND SUPPORT VECTOR CLUSTERING OF CATEGORICAL DATA

Marina Marino, Cristina Tortora

Dipartimento di Matematica e Statistica
Università degli Studi di Napoli Federico II, Italia
mari@unina.it; cristina.tortora@unina.it

Abstract

Standard clustering methods fail when data are characterized by non-linear associations. A suitable solution consists in mapping data in a higher dimensional feature space where clusters are separable. The aim of the present contribution is to propose a new technique in this context and to compare it with k-means technique.

Keywords: kernel method, machine learning, support vector clustering.

1. INTRODUCTION

Categorical data arise often in many fields, including biometrics, economics, management, manufacturing, marketing, psychology, and sociology. So the use of statistical methods for categorical data is ever increasing in today's world.

In this paper we focus on clustering methods for categorical data. Cluster analysis techniques aim at organizing information about variables so that relatively homogeneous groups, or "clusters" can be formed. The clusters formed with this family of methods should be highly internally homogenous (members are similar to one another) and highly externally heterogenous (members are not like members of other clusters). In other words, clustering algorithms aim at finding homogeneous groups with respect to their association structure among variables. When there is linear association between variables, suitable transformations of the original variables or proper distance measures make it possible to get satisfactory solutions (Saporta 1990). However when data are characterized by non-linear association the actual cluster structure remains invisible to these approaches. This is the case of categorical data that are characterized by non-linear association because this type of data can be combined forming a limited subspace of data space. This means that categorical data are characterized by clusters of

arbitrary shape and/or nested while classical cluster analysis techniques are able to find out clusters of spherical shape and not nested. To overcome this problem a suitable solution is to project data into a higher dimensional space (feature space). However, when the number of variables is large, projecting data into a higher dimensional space is a self-defeating and computationally unfeasible task. So a Factor Analysis on the raw data matrix allows to transform categorical data into continuous one and to reduce the dimensionality of the problem.

In this paper we focus on large categorical data set and propose a clustering approach based on a multistep strategy: *i*) Factor Analysis on the raw data matrix; *ii*) projection of the first factor coordinates into a higher dimensional space; *iii*) clusters identification in the high dimensional space; *iv*) clusters visualization in the factorial space.

In the following the synergic advantage of this mixed strategy is motivated and a comparison of results obtained using both our strategy and one of the most used clustering method, k-means, on the first factor coordinates is carried out.

2. SUPPORT VECTOR CLUSTERING ON MCA FACTORS

2.1 FACTOR ANALYSIS ON CATEGORICAL DATA

As said before, when the number of variables is large, projecting data into a higher dimensional space is a self-defeating and computationally unfeasible task. In order to carry only the significant association structure in the analysis, dealing with continuos variables, some authors propose to perform a Principal Component Analysis on the raw data, and then to project the first components in a higher dimensional feature space (Ben-Hur et al. 2002). In the case of categorical variables, the dimensionality depends on the whole number of categories, this implies an even more dramatic problem of sparseness. Moreover, as modalities are a finite number, the association between variables is non-linear.

Multiple Correspondence Analysis (MCA) on the raw data matrix permits to combine the categorical variables into continuos variables that preserve the non-linear association structure and to reduce the number of variables, dealing with sparseness few factorial axes can represent a great part of the variability of the data. It is well known that inertia rate (percentage of variability explained by each factor) is a pessimistic measure of factors explicative power, due to disjunctive coding. So we use the corrective factor proposed by Benzécri (Benzécri 1979) to determinate the effectiveness explained variability of first factors. Let us indicate with $\mathbf{Y}$ the $n \times q$ coordinates matrix of n points into the orthogonal space spanned

by the first q MCA factors. For sake of brevity we do not go into the MCA details; interested readers are referred to Greenacre's book (Greenacre 2007). Mapping the first factorial coordinates into a feature space permits to cluster data via a Support Vector Clustering approach (Marino et al.).

2.2 SUPPORT VECTOR CLUSTERING

Support Vector Clustering (SVC) is a non parametric cluster method based on support vector machine that maps data points from the original variable space to a higher dimensional feature space trough a proper kernel function (Muller et al. 2001).

A feature space is an abstract t-dimensional space where each statistical unit is represented as a point. Given an observations $\times$ variables data matrix $\mathbf{X}$ with general term x_{ij} and $i = 1,2,\ldots,n$ and $j = 1,2,\ldots,p$, any generic row or column vector of $\mathbf{X}$ can be represented into a feature space using a non linear mapping function. Formally, the generic column (row) vector $\mathbf{x}_j$ ($\mathbf{x}_i'$) of $\mathbf{X}$ is mapped into a higher dimensional space F trough a function

$$\varphi(\mathbf{x}_j) = (\phi_1(\mathbf{x}_j), \phi_2(\mathbf{x}_j), \ldots, \phi_t(\mathbf{x}_j)),$$

with $t > p$ ($t > n$ in the case of row vectors) and $t \in \mathbb{N}$.

The solution of the problem implies the identification of the minimal radius hypersphere that includes the images of all data points, points that are on the surface of the hypersphere are called *support vectors*. In the data space the support vectors divide the data in clusters. The problem is to minimize the radius under the constraint that all the points belong to the hypersphere: $r^2 \geq \left\|\varphi(\mathbf{x}_j) - \mathbf{a}\right\|^2 \quad \forall j$, where $\mathbf{a}$ is the center of the hypersphere and $\|\cdot\|$ denotes the Euclidean norm.

To avoid that only the most far point determines the solution, slack variables $\xi_j \geq 0$ can be added:

$$r^2 + \xi_j \geq \left\|\varphi(\mathbf{x}_j) - \mathbf{a}\right\|^2 \quad \forall j.$$

This problem can be solved by introducing the Lagrangian:

$$L(r, \mathbf{a}, \xi_j) = r^2 - \sum_j (r^2 + \xi_j - \left\|\varphi(\mathbf{x}_j) - \mathbf{a}\right\|^2)\beta_j - \sum_j \xi_j \mu_j + C\sum_j \xi_j, \qquad (1)$$

where $\beta_j \geq 0$ and $\mu_j \geq 0$ are Lagrange multipliers, C is a constant and $C\sum_j \xi_j$ is a penalty term. To solve the minimization problem we set to zero the derivate of L with respect to r, $\mathbf{a}$ and ξ_j and we get the following solutions:

$$\sum_j \beta_j = 1$$

$$\begin{aligned} \mathbf{a} &= \textstyle\sum_j \beta_j \varphi(\mathbf{x}_j) \\ \beta_j &= C - \mu_j \end{aligned}$$

We remind that Karush-Kuhn-Tucker complementary condition implies:

$$\begin{aligned} &\xi_j \mu_j = 0 \\ &(r^2 + \xi_j - \left\|\varphi(\mathbf{x}_j) - \mathbf{a}\right\|^2)\beta_j = 0 \end{aligned}$$

The Lagrangian is a function of r, $\mathbf{a}$ and μ_j. Turning the Lagrangian into the more simple Wolfe dual form, which is a function of the variables β_j, we obtain:

$$W = \sum_j \varphi(\mathbf{x}_j)^2 \beta_j - \sum_{j,j'} \beta_j \beta_{j'} \varphi(\mathbf{x}_j) \cdot \varphi(\mathbf{x}_{j'}), \tag{2}$$

with the constraints $0 \leq \beta_j \leq C \quad \forall\{j, j'\}$.
It is worth noticing that in (2) the function $\varphi(\cdot)$ only appear in products. The dot products $\varphi(\mathbf{x}_j) \cdot \varphi(\mathbf{x}_{j'})$ can be computed using an appropriate kernel function $K(\mathbf{x}_j, \mathbf{x}_{j'})$. The Lagrangian W is now written as:

$$W = \sum_j K(\mathbf{x}_j, \mathbf{x}_j)\beta_j - \sum_{j,j'} \beta_j \beta_{j'} K(\mathbf{x}_j, \mathbf{x}_{j'}). \tag{3}$$

The SVC problem requires the choice of a kernel function and a respective suitable parameter α. The choice of the kernel function remains a still open issue (Cristianini, Shawe-Taylor 2006).
There are several proposal in the recent literature: *Linear Kernel* ($k(x_i, x_j) = \langle x_i \cdot x_j \rangle$), *Gaussian Kernel* ($k(x_i, x_j) = exp(-q \left\|x_i - x_j\right\|^2 / 2\sigma^2)$) and *polynomial Kernel* ($k(x_i, x_j) = (\langle x_i \cdot x_j \rangle + 1)^d$ with $d \in N$ and $d \neq 0$) are among the most largely used functions. In the present work we adopt a polynomial kernel function; the choice was based on the empirical comparison of the results (Abe 2005).

Most important for the final clustering result is the choice of the parameter α because this parameter affects the number of clusters.

To simplify the notation, we indicate with $K^*(\cdot)$ the parametrised kernel function: then in our specific context the problem consists in maximizing the following quantity with respect to β

$$W = \sum_m K^*(\mathbf{y}_m, \mathbf{y}_m)\beta_m - \sum_{m,m'} \beta_j \beta_{m'} K^*(\mathbf{y}_m, \mathbf{y}_{m'}), \tag{4}$$

where $\mathbf{y}_m$ represents the generic coordinate obtained via MCA, $1 \leq m \leq q$.

The distance of the image of each point in the features space and the center of the hypersphere is:

$$R^2(y) = \|\varphi(\mathbf{y}) - \mathbf{a}\|^2 \tag{5}$$

Applying previous results the distance is obtained as:

$$R^2(\mathbf{y}) = K^*(\mathbf{y},\mathbf{y}) - 2\sum_j K^*(\mathbf{y}_j,\mathbf{y})\beta_j + \sum_{j,j'} \beta_j \beta_{j'} K^*(\mathbf{y}_j,\mathbf{y}_{j'}). \tag{6}$$

Points, whose distance from the surface of the hypersphere is less then ξ, are the support vectors and they define a partition of the feature space. For these point, $0 < \beta_i < C$, points with $\beta_i = C$ are called bounded support vectors and they are outside the feature-space hypersphere. If $\beta_i = 0$ the point is inside the feature-space hypersphere. The number of support vectors affects the number of clusters, as the number of support vectors increases the number of clusters increases. The number of support vectors depends on α, as α increases the number of support vectors increases because the contours of the hypersphere fit better the data, as C decreases the number of bounded support vectors increases and their influence on the shape of the cluster contour decreases.

The (squared) radius of the hypersphere is:

$$r^2 = \{R(y_i)^2 | y_i \text{ is a support vector}\} \tag{7}$$

2.3 CONE CLUSTER LABELING

The last clustering phase consists in assigning the points projected in the feature space to the classes.

It is worth reminding that the analytic form of $\{\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \ldots, \phi_t(\mathbf{x})\}$ is unknown, so that computing points coordinates in the feature space is an unfeasible task. Alternative approaches permit to define the points memberships without computing all coordinates. In this paper, in order to assign points to clusters we use the *cone cluster labeling algorithm* (Lee, Daniels 2006) adapted to the case of polynomial kernel.

The cone cluster labeling (CCL) is different from the classical methods because it is not based on the distances between pairs of point. This method look for a surface that cover the hypersphere, this surface consists of a union of coned-shaped regions. Each region is associated with a support vector's features space image, the phase of each cone $\Phi_i = \angle(\phi(v_i)O\mathbf{a})$ is the same, where v_i is a support vector, $\mathbf{a}$ is the center of the minimal hypersphere and O is the feature space

origin. The image of each cone in the data space is an hypersphere, if two hypersphere overlap the two support vectors belong to the same class. So the object is to find the radius of these hypersphere in the data space $\|v_i - g_i\|$ where g is a generic point of the surface of the hypersphere. It can be demonstrated that $\mathbf{K}(v_i, g_i) = \sqrt{1-r^2}$ (Lee, Danels 2006) so in case of polynomial kernel we obtain:

$$\begin{aligned} \mathbf{K}(v_i, g_i) &= ((v_i g_i') + 1)^d \\ \sqrt{1-r^2} &= ((v_i g_i') + 1)^d. \end{aligned} \tag{8}$$

Starting from the (8) we can compute the coordinate of g_i: $g_i' = [(1-r^2)^{\frac{1}{2d}} - 1]v_i'$ and consequently the value of $\|v_i - g_i\|$. If the distances between two generic support vector is less then the sum of the two radius they belong to different cluster.

The computational cost of the CCL method is $O(N_{SV}^2)$ while the computational cost of the classical method, complete graph (CG), is $O(N^2)$, where N_{SV} is the number of support vectors and N is the number of units. When the number of support vectors is small CCL is faster than CG.

3. EMPIRICAL EVIDENCE

The method has been applied to the *Mushrooms* datasets. The access information is available from the UCI Machine Learning Repository web site[1]. The dataset is composed by 22 variables, 21 variables represent characteristics of mushrooms and the last one is a binary variable that divide poisonous and edible mushrooms. Each item is a mushroom. The variable stalk root have been discarded because it contains a lots of missing values, so that the final dataset is made up of 7857 observations and 21 categorical variables.

In order to reduce the dimensionality of the problem, at the first step, a MCA algorithm with the Benzécri corrective factor have been applied. We can see that, starting from the fifth factor the growth of the explained variability is minimal (fig.1), this is due to the sparseness of the data. Starting from the fifth factor, the others represent only ground noise present in the data, for this reason we computed the coordinates of the units on the first four MCA factors that explain 80% of the variability.

In the second step we apply a SVC algorithm, this requires the choice of a kernel function.With a polynomial kernel with parameters $d = 1$ and $C = 0.01$

[1] *http://archive.ics.uci.edu/*

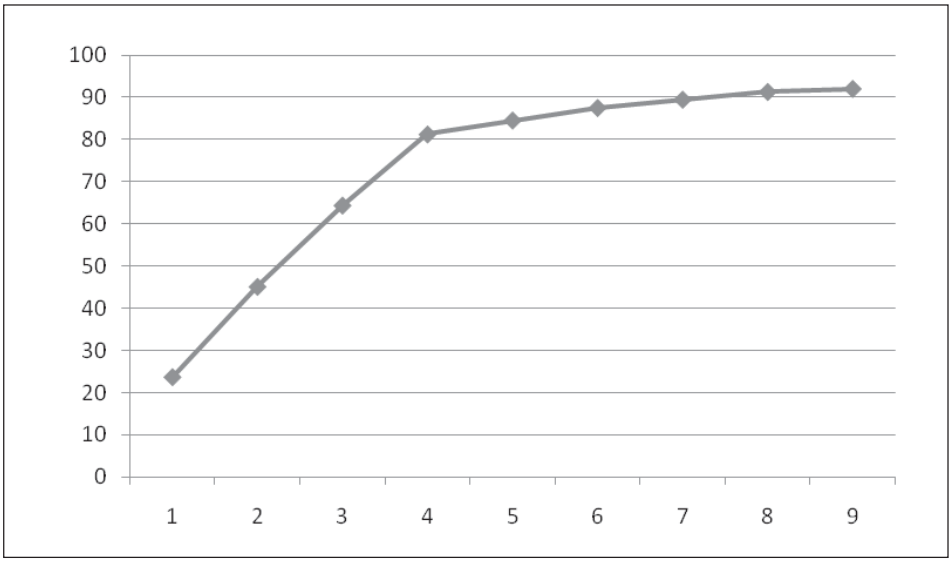

Fig. 1: Explained variability of first four MCA factor.

we obtain five classes. With different values of parameters d and C we obtain always four classes. Using a Gaussian kernel we can obtain different structure as parameters vary, that is to say that changing parameters we can obtain different number of clusters. The best results obtained are with parameters $d = 9$ and $C = 0.01$, in this case data are grouped in five classes.

In order to evaluate the better performing procedure we use the CATANOVA method (Singh 1993). This method is analogous to ANOVA method for the case of categorical data. The CATANOVA method tests the null hypothesis $H_0 : p_{ij} = p_i \quad \forall\, i = 1,...,p$ and $\forall\, j = 1,...,k$, where p is the number of variables and k the number of clusters. The null hypothesis is rejected using both polynomial or Gaussian kernel. The higher value of CATANOVA index (CA) is $CA = 8.073e^5$ reached when the data are grouped in five clusters. This clustering is shown in figure 2.

On same MCA factors a k-means algorithm have been applied. K-means is an iterative method that find the structure that minimize the within variance of clusters. The solution is not stable because the algorithm can run into local minima, so reiterating the algorithm we can obtain different solutions. For this reason we replicated the method 1000 times, we use CATANOVA to evaluate the solutions obtained. In the best case $CA = 8.284e^5$, higher than the one obtained using SVC, and the within variance is $4.055e^3$, lower than using SVC. However

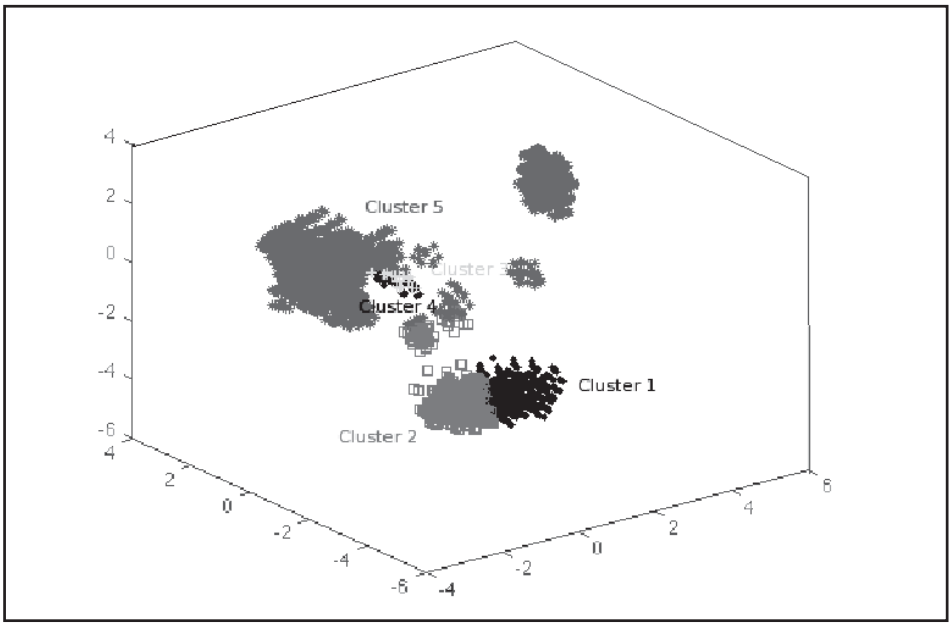

Fig. 2: Clusters obtained with SVC algorithm.

this happens in less than 1% of cases only. In the most frequent case we have $CA = 8.039e^5$, lower than the one obtained with SVC, and the within variance is $5.925e^3$, higher than using SVC. Observing figure 3 and reminding that using SVC the value of CA is $8.073e^5$, we note that SVC performs better than k-means in 95% of cases.

In the dataset there is a variable that have not be included in the analysis, it divides poisonous and edible mushrooms. We have analyzed the structure of the cluster (fig.2) with respect to this variable:

- Cluster 1 contains 1296 poisonous mushrooms;
- Cluster 2 is made by 31 poisonous mushrooms;
- Cluster 3 is composed by 1638 poisonous and 40 edible mushrooms;
- Cluster 4 contains 120 edible mushrooms;
- Cluster 5 is made by 822 poisonous and 3910 edible mushrooms.

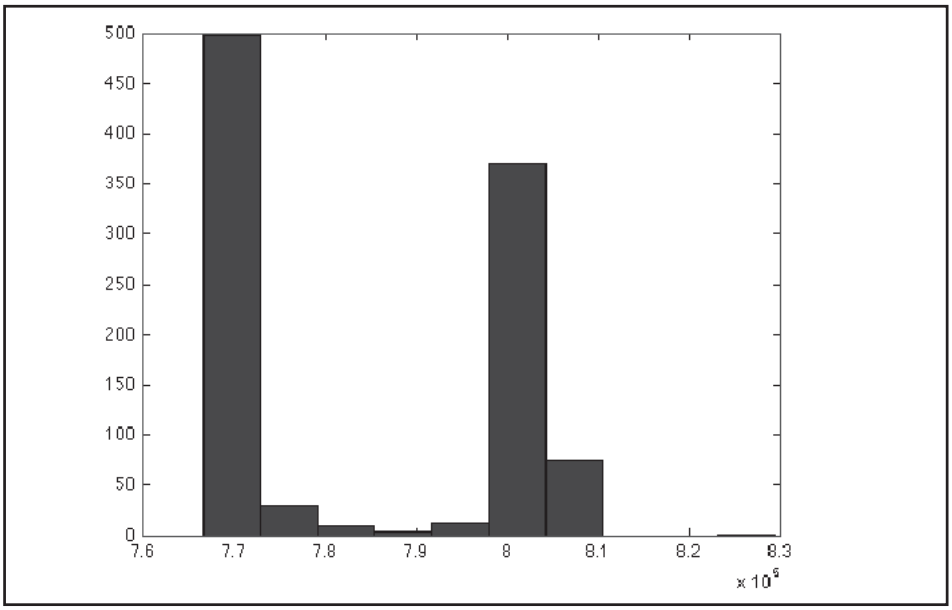

Fig. 3: C on 1000 iterations of k-means algorithm.

Three clusters are homogeneous. Cluster number 3 contains a small number of poisonous mushrooms (2%). Only the cluster number 5 is heterogeneous with 17% of poisonous mushrooms. Using k-means algorithm clusters structure change, in the best case we obtain (fig.4):

- Cluster 1 contains 865 poisonous mushrooms;
- Cluster 2 contains 65 edible mushrooms;
- Cluster 3 is composed by 750 poisonous and 40 edible mushrooms;
- Cluster 4 is made by 55 poisonous mushrooms;
- Cluster 5 is made by 2172 poisonous and 3910 edible mushrooms.

Three clusters are homogeneous, cluster number 3 is more heterogeneous than cluster 3 obtained with SVC (5%). Cluster number 5 have a great heterogeneity whit 35% of poisonous mushrooms.

In the most frequent case using k-means we obtain (fig.5):

- Cluster 1 contains 1296 poisonous mushrooms;

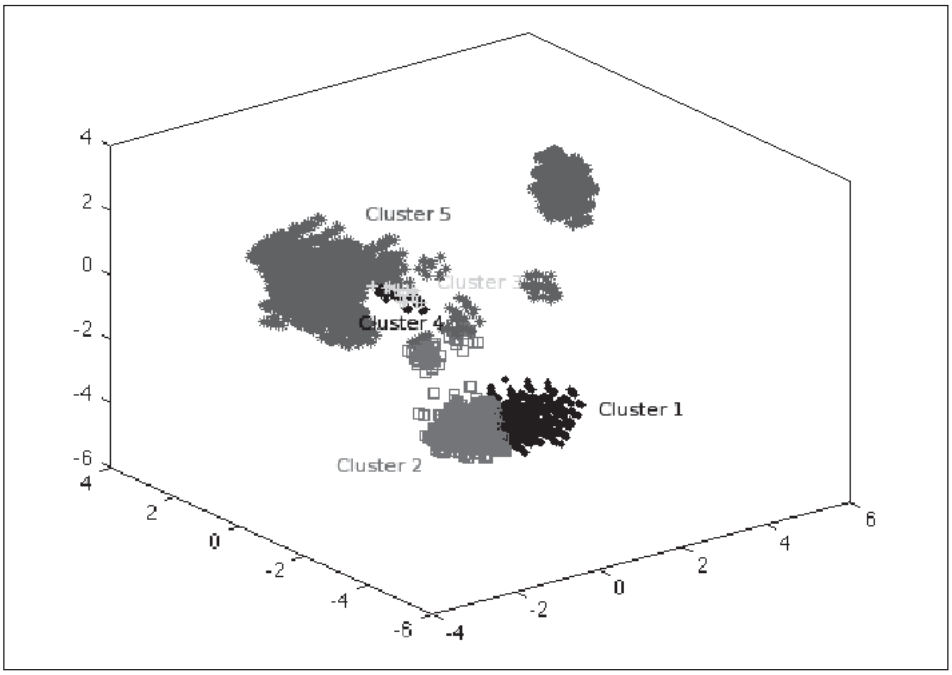

Fig. 4: Clustering obtained in the best case with k-means.

- Cluster 2 is composed by 160 poisonous and 217 edible mushrooms;
- Cluster 3 contains 120 edible mushrooms;
- Cluster 4 is made by 718 poisonous and 1919 edible mushrooms;
- Cluster 5 is made by 160 poisonous and 2017 edible mushrooms.

In this case only two clusters are homogeneous, number 1 and 3. This cluster are the same obtained using SVC. The other clusters are all heterogeneous. Thus using SVC we obtain clusters homogeneous with respect to the poisonous/edible variable.

4. CONCLUSION

Dealing with large data-sets of categorical data the method proposed can be an alternative to traditional clustering methods. The first step consists in a reduction of

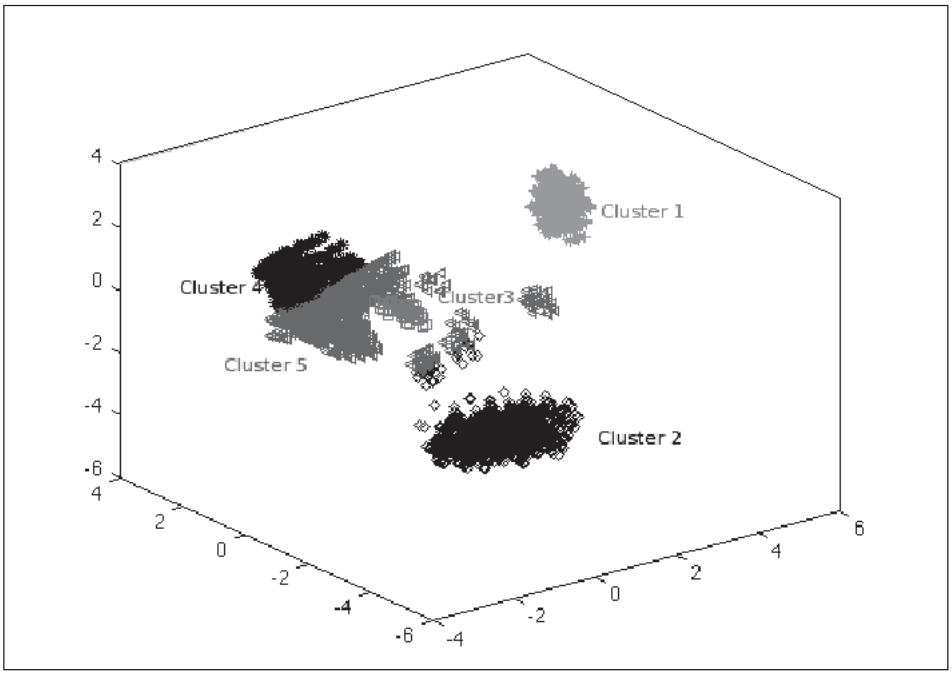

Fig. 5: Clustering obtained in the most frequent case with k-means.

dimensionality without losing nonlinear relations between variables by means of MCA, in such a way we obtain also a quantification of categorical data. Successively SVC method was applied on the MCA factors and clusters are obtained. To label cluster we used cone cluster labeling method adapted for polynomial kernel, too.

Main advantages of this strategy are that the classes can be seen on a factorial axes and it can help in the interpretation of the results. Moreover SVC algorithm converges to a global optimal solution. Comparing SVC and k-means we can say that in the best case k-means performs better than SVC but it happens only in 1% of cases. If we look at the poisonous/edible variable, SVC forms more homogeneous clusters than k-means.

There are still some open issues: the choice of the kernel function is done empirically, it does not exist an analytic way to choose it; the number of classes depends on the kernel parameters so it can not be chosen directly.

Future works will be in these directions.

REFERENCES

ABE, S. (2005). *Support vector machine for pattern classification*, Springer.

BEN-HUR, A., HORN, D., SIEGELMANN, H. T., VAPNIK, V. (2002). A support vector method for clustering, *Advances in neural network processing system*: 367-373 .

BENZECRI, J. P. (1979). Sur le calcul des taux d'inertie dans l'analyse d'un questionnaire, *Le Cahiers de l'Analyse des Données* **4**(4): 377–378.

CRISTIANINI, N., SHAWE-TAYLOR, J. (2006). *An introduction to Support Vector Machine*, Cambridge University press.

GREENACRE, M. J. (2007). *Correspondence Analysis in Practice, second edition*, Chapman and Hall/CR.

LEE, S.-H., DANIELS, K. M. (2006). Cone cluster labeling for support vector clustering, *Proceeding of the sixth SIAM international conference on data mining*, Bethesda: 484-488.

MARINO, M., PALUMBO, F., TORTORA, C. (submitted). Clustering in feature space for interesting pattern identification of categorical data, *extended-paper proceedings of the SIS 2009 statistical conference on Statistical Methods for the analysis of large data-sets*, Pescara.

MULLER, K. R., MIKA, S., RATSCH, G.,TSUDA, K., SCHOLKOPF, B. (2001). An introduction to kernel-based learning algorithms, *IEEE transaction on neural networks* **12**: 181-201.

SAPORTA, G. (1990). *Probabilités Analyse des Données et Statistiques*, Technip, Paris.

SINGH, B. (1993). On the analysis of variance method for nominal data, *Sankhyā: The Indian Journal of Statistics, Series B* **55**: 40-47.

UN CONFRONTO TRA K-MEANS E SUPPORT VECTOR CLUSTERING SU DATI CATEGORICI

Riassunto

I metodi di classificazione automatica classici falliscono quando i dati sono caratterizzati da relazioni non lineari, una possibile soluzione consiste nel proiettare i dati in un feature space di dimensioni maggiori dove le classi sono separabili. L'obiettivo di questo articolo é di proporre una nuova tecnica in questo ambito e di confrontarla con la tecnica di classificazione k-means.