# Clustering in the Membership Embedding Space

Maurizio Filippone [a], Francesco Masulli [b,c,*], Stefano Rovetta [b]

[a] *Department of Computer Science, University of Sheffield, Regent Court*
*211 Portobello Street Sheffield, S1 4DP, United Kingdom.*

[b] *DISI, Dipartimento di Informatica e Scienze dell' Informazione,*
*Università di Genova, and CNISM, Via Dodecaneso 35, Genoa, Italy.*

[c] *Center for Biotechnology, Temple University*
*1900 N 12th Street Philadelphia, PA 19122, USA.*

**Abstract**

In several applications of data mining to high-dimensional data, clustering techniques developed for low-to-moderate sized problems obtain unsatisfactory results. This is an aspect of the *curse of dimensionality* issue. A traditional approach is based on representing the data in a suitable similarity space instead of the original high-dimensional attribute space. In this paper, we propose a solution to this problem using the projection of data onto a so-called Membership Embedding Space obtained by using the memberships of data points on fuzzy sets centered on some prototypes. This approach can increase the efficiency of the popular Fuzzy C-Means method in the presence of high-dimensional data sets, as we show in an experimental comparisons. We also present a constructive method for prototypes selection based on simulated annealing that is viable for semi-supervised clustering problems.

*Key words:* High Dimensional Data Sets; Unsupervised Clustering; Supervised Clustering; Fuzzy Sets; Embedding Spaces; Fuzzy C-Means; Simulated Annealing.

## 1 Introduction

Clustering methods are useful tools for data mining. They can be employed both in an *unsupervised* way, when available data are unlabeled (or available labels are unreliable, or when the data labeling task is too expensive), and in a *semi-supervised* way when a small amount of knowledge is available concerning either pairwise

[*] Corresponding author.
*Email addresses:* filippone@dcs.shef.ac.uk (Maurizio Filippone), masulli@disi.unige.it (Francesco Masulli), rovetta@disi.unige.it (Stefano Rovetta).

(must-link or cannot-link) constraints between data items or class labels for some items.

Unfortunately, in several problems of data mining, data lie in a very high dimensional space (thousands of dimensions). In these situations, the direct application of clustering algorithms developed for low-dimensional spaces (e.g., K-means [Steinhaus (1956), Lloyd (1982)], BIRCH [Zhang et al. (1996)], CURE [Guha et al. (1998)], CLARANS [Ng & Han (2002)], etc.) often leads to poor results. Even after feature selection, one may be left with hundreds of dimensions (and further reductions will significantly degrade the results). Even techniques designed for large quantities of data, such as DBScan [Ester et al. (1996)], only focus on the problem of large-cardinality datasets, thus effectively making the assumption of (relatively) small dimensionality. This is an aspect of the well-known *curse of dimensionality* issue [Bellman (1961)].

Many clustering algorithms suffer from being applied in high-dimensional spaces, as clustering algorithms often seek for areas where data are dense. Sometimes the cardinality of the data sets available is even less than the number of variables, as in the case of the analysis of many bioinformatics data sets or in web mining problems. This means that data span only a subspace within the data space. In these conditions, it is not easy to define the concept of volumetric density.

Moreover, when space dimensionality is high or even moderate (as low as 10-15), the distance of a point to its farthest neighbor and to its nearest neighbor tend to become equal [Beyer et al. (1999), Aggarwal & Yu (2002)]. Therefore the evaluation of distances, and the concept of *nearest neighbor* itself, become less and less meaningful with growing dimensions. Defining clusters on the basis of distance measures requires that distances can be estimated. For instance, one of the most commonly used methods, K-means clustering [Steinhaus (1956), Lloyd (1982)], is based on iteratively computing distances and cluster averages. Increasing the data space dimensionality may introduce a large number of suboptimal solutions (local minima), and the nearest-neighbor criterion which is the basis of the method may even become useless. This problem is not avoided even when K-means is modified in the direction of incorporating fuzzy concepts, e.g. as for the FCM (Fuzzy C-Means) algorithm [Dunn (1973), Bezdek (1981)].

A possible approach alleviating these problems is based on representing the data in a suitable similarity space instead of the original high-dimensional attribute space (see e.g. Strehl & Ghosh (2003), Filippone et al. (2008), and Filippone (2009)).

In this paper, we propose a solution to the highlighted problems using the projection of data onto a so-called Membership Embedding Space (MES). Such projection is obtained by using the memberships of data points on fuzzy sets centered on some prototypes selected among data points themselves. We will demonstrate that this approach can increase clustering efficiency of the popular Fuzzy C-Means (FCM) [Bezdek (1981)] algorithm in the presence of high-dimensional data sets. To this aim, we will experimentally compare the performances of FCM in the original data space, with those in the Distance Embedding Space (following the approach proposed by Pękalska and Duin [Pekalska et al. (2001)]) and Membership Embedding Space, using different prototypes-data ratios. Moreover, we will present a constructive method for prototypes selection based on simulated annealing that is viable for semi-supervised clustering problems as well.

In Section 2 a fuzzy embedding for high dimensional data sets, is presented; in Section 3 we recall the main aspects of Fuzzy C-Means clustering algorithm, and in Section 4 we present a constructive approach for selecting an optimal set of prototypes in the fuzzy embedding. The experimental results are reported in Section 5. The conclusions are given in Section 6.

## 2 Membership Embedding Space

A notable complexity reduction of data mining problems in the presence of large-dimensional data sets can be provided by representations in a similarity space or embedding space based on an assigned pairwise similarity (or dissimilarity) transformation (see e.g. Strehl & Ghosh (2003), Filippone et al. (2008), and Filippone (2009)).

Given a data set $X$ of cardinality $n$, $X = \{x_1, x_2, \ldots, x_n\}$ in a $d$ dimensional space, the (dis-)similarity transformation $v(x_i, x_j)$ maps the $n \times d$ data matrix into a more dense symmetric $n \times n$ matrix of similarities $v$, with $v_{ik} = v(x_i, x_k) \;\; \forall i, k$.

Mutual distances or other pairwise pattern evaluation methods such as kernels [Shawe-Taylor & Cristianini (2004)] may be used as (dis-)similarity transformations. If the cardinality of the data set is small compared to the input space dimensionality, data sets can be represented in the embedding space in a very compact way.

Applications of projection onto (dis-)similarity embedding spaces to clustering are reported, e.g. in Fred & Leitão (2003), and Rovetta & Masulli (2006). Pekalska et al. (2001) developed a set of methods based on representing each pattern according to a set of similarity measurements with respect to other patterns in the data set. As they pointed out, the (dis-)similarity measure should be a metric, since metrics preserve the *reverse of the compactness hypothesis* [Pekalska et al. (2001)]: "objects that are similar in their representation are also similar in reality and belong, thereby, to the same class".

Often non-metric distances are used as well. Moreover, sometime the (dis-)similarity matrix can be reduced from a square matrix $n \times n$ to a smaller rectangular matrix $n \times s$, by selecting $s \leq n$ reference points (called *prototypes*) and computing the (dis-)similarities of the data with them. If the embedding dimension $s$ is small compared with $d$ (i.e. $s/d \ll 1$), some points could have an ambiguous representation, In order to avoid the previously highlighted problems, in this paper we study an embedding based on the space of memberships to fuzzy sets [Zadeh (1965)] centered on selected prototypes.

The memberships to fuzzy sets centered on the prototypes are modeled using the following normalized function:

$$v_{ik} = \frac{\exp\left[-\beta d_{i,k}^2\right]}{\sum_l \exp\left[-\beta d_{l,k}^2\right]} \quad \beta \in \Re^+, \tag{1}$$

where the parameter $\beta$ regulates the spread of the membership function. The denominator normalizes the sum of the memberships to the prototypes to sum up to one. The matrix $\mathbf{V} = [v_{ik}]$ is the similarity matrix. Note that the $\mathbf{V}$ is rectangular, since we select a number of prototypes $s \leq n$.

In this way, in the Membership Embedding Space (MES) a data point $x_i$ is represented as a row of $v$, i.e., $x_i = (v_{i1}, v_{i2}, \ldots, v_{in})$. Due to the localized definition of fuzzy sets, this vector of memberships contains only few non-null elements, in correspondence of the nearest prototypes in the original data space. If the spread of membership is large (i.e., large $\beta$) many of these elements are non-null, otherwise for $\beta$ going to zero, only the data points corresponding to the selected prototypes have at least one non-null element.

The results of a clustering method will be affected by the number and the positions of the prototypes as well as by the value of spread parameter $\beta$. Placing the prototypes is a combinatorial search problem which will be tackled by a Simulated Annealing approach.

## 3 Fuzzy C-Means Algorithm

In the experiments reported in this paper, we have used the Fuzzy C-Means (FCM) algorithm [Bezdek (1981)] as the clustering algorithm. Other clustering techniques can be applied, but we focus on a single choice for the sake of clarity.

The FCM algorithm performs the minimization of the following functional:

$$J_m(\mathbf{U}, Y) \equiv \sum_{i=1}^{n} \sum_{k=1}^{c} (u_{ik})^m d_{ik} \tag{2}$$

where: $X = \{x_1, x_2, \ldots, x_n\}$ is a data set containing $n$ unlabeled sample points; $Y = \{y_1, y_2, \ldots, y_c\}$ is the set of the centers of clusters; $\mathbf{U} = [u_{ik}]$ is the $c \times n$ fuzzy c-partition matrix, containing the membership values of all samples to all prototypes; $m \in (1, \infty)$ is the fuzziness control parameter; $d_{ik}$ is a dissimilarity measure between data point $x_i$ and the center $y_k$ of a specific cluster $k$. In the rest of this paper we will use the Euclidean squared distance $d_{ik} \equiv ||x_i - y_k||^2$ as the dissimilarity measure.

The clustering problem can be formulated as the minimization of $J_m$ with respect to $Y$, under the normalization constraint $\sum_{k=1}^{c} u_{ik} = 1$.

The necessary conditions for minimizing $J_m$ are then:

$$y_k = \frac{\sum_{i=1}^{n} (u_{ik})^m x_i}{\sum_{i=1}^{n} (u_{ik})^m} \qquad \text{for all } k, \tag{3}$$

$$u_{ik} = \left[ \sum_{j=1}^{c} \left( \frac{d_{ik}}{d_{jk}} \right)^{\frac{1}{m-1}} \right]^{-1} \qquad \text{for all } i, k. \tag{4}$$

The Fuzzy C-Means algorithm usually starts with a random initialization of the fuzzy c-partition matrix **U** or of the centroids $y_k$. Then, it iterates Eq.s 3 and 4 until convergence. Usually, the convergence is checked by comparing the change in the position of the centroids or in the cost function with some fixed thresholds.

Note that in the limit for $m \rightarrow 1$ the fuzzy C-Means Functional $J_m$ (Eq. (2)) tends to the expectation of the K-Means global error $< E > \equiv \sum_{i=1}^{n} \sum_{k=1}^{c} u_{ik} d_{ik}$, and the FCM behaves as the classic K-means (or Hard C-means) algorithm [Steinhaus (1956), Lloyd (1982), Duda & Hart (1973)].

## 4 Simulated Annealing Prototype Selection Algorithm

As already noted, the selection of the optimal set of prototypes for constructing the Membership Embedding Space is a combinatorial search problem. A constructive heuristic algorithm able to select the set of prototypes leading to (sub-)optimal clustering in the MES can be based on Simulated Annealing (SA) [Kirkpatrick et al. (1983), Černý (1985)] that is a global search probabilistic technique inspired to annealing in metallurgy,

“Physical” annealing as used in metallurgy involves heating a material, and then cooling it slowly and in a controlled fashion. The aim of this process is to allow the crystal lattice to reorganize so as to reduce the defects and to reach a more stable, and therefore stronger, inner structure. Heating allows atoms to detach from their initial positions (corresponding to a local minimum of the internal energy) and to float randomly through states of higher energy; slow cooling allows them more chances to find configurations with internal energy lower than the initial one.

SA is an adaptation of the Metropolis-Hastings algorithm [Metropolis et al. (1953)] aimed to simulate the behavior and small fluctuations of a system of atoms starting from an initial configuration, by the generation of a sequence of iterations. In the Metropolis algorithm each iteration comprises a random perturbation (modification) of the actual configuration (state vector) and the computation of the corresponding energy variation ($\Delta E$). If $\Delta E < 0$ the transition is unconditionally accepted, otherwise the transition is accepted with probability given by the Boltzmann distribution:

$$P(\Delta E) = \exp\left(\frac{-\Delta E}{KT}\right) \tag{5}$$

where K is the Boltzmann constant and T the temperature.

In SA this approach is generalized to the solution of general optimization problems [Kirkpatrick et al. (1983)], by using an *ad hoc* selected cost function (*generalized energy*), instead of the physical energy; therefore, it can also be employed when the search space is discrete, as in combinatorial search problems. SA works as a probabilistic hill-climbing procedure searching for the global optimum of the cost function [Romeo (1986)]. $K$ is usually set to 1, while the temperature $T$ controls the size of the search area, and is gradually lowered until no further improvements of the cost function are noticed. SA can work in very high-dimensional searches, given enough computational resources. In applications, it is important to trade-off the quality of the solution and the computational cost; a slower decreasing of the temperature allows the system to reach better solutions but more time is required to explore the state space.

In Tab. 1 the proposed *Simulated Annealing Prototypes Selection* (SA-PS) algorithm is shown. In our approach the state of the system is represented by a binary mask $\mathbf{g} = (g_1, g_2, \ldots, g_n)$, where each bit $g_i$ (with $i = 1, \ldots, n$) corresponds to the selection ($g_i = 1$) / deselection ($g_i = 0$) of a prototype. The initialization of the vector mask $\mathbf{g}$ (Step 2) is done by generating $s_0$ integer numbers with uniform distribution in the interval $[1, n]$ and setting the corresponding bits of $\mathbf{g}$ to 1 and the remaining ones to 0. At each step only $s$ prototypes are selected from the original set of $n$ patterns. A perturbation or move is done in the following way: (1) chose randomly $w \in [w_{\min}, w_{\max}]$ and $v \in [v_{\min}, v_{\max}]$; (2) $w$ bits of $\mathbf{g}$ set to 1 are switched to 0; (3) $v$ bits of $\mathbf{g}$ set to 0 are switched to 1.

The values $w_{\min}, w_{\max}, v_{\min}, v_{\max}$ can be used to reduce or to increase the variability of each perturbation.

Once a set of prototypes is selected, it is possible to represent each pattern in the Membership Embedding Space (MES) and perform clustering.

The generalized energy $E$ is computed as a linear combination between an assigned clustering quality measure $\varepsilon$ and the number of selected prototypes $s$:

$$E = \varepsilon + \lambda s \tag{6}$$

The clustering quality measure $\varepsilon$ can be a function of either the cost function as-

Table 1
Simulated Annealing Prototype Selection (SA-PS) Algorithm.

```
(1) Initialize parameters (see list in Tab. 3);
(2) Initialize the binary mask g at random;
(3) Perform clustering and evaluate the generalized system energy $E$;
(4) do
(5) Initialize $f = 0$ (number of iterations), $h$=0 (number of successes);
    (a) do
    (b) Increment number of iterations $f$;
    (c) Perturb mask g;
    (d) Perform clustering and evaluate the generalized system energy $E$;
    (e) Generate a random number $rnd$ in the interval [0,1];
    (f) if $rnd < P(\Delta E)$ then
        (i) Accept the new g mask;
        (ii) Increment the number of successes $h$;
    (g) endif
    (h) loop until $h \leq h_{\min}$ and $f \leq f_{\max}$;
(6) update $T = \alpha T$;
(7) loop until $h > 0$;
(8) end.
```

sociated to the clustering algorithm, a clustering validation index, or, in the case of *semi-supervised* clustering (where we have a partially labeled data set), the *Representation Error* (RE). RE is the number of data points in each cluster disagreeing with the majority label in that cluster, summed over all clusters and expressed as a percentage.

Note that the introduction of the number of selected prototypes $s$ in the computation of $E$ penalizes situations in which the number of selected prototypes is high, effectively resulting as a complexity penalty term. This choice of $E$ leads to the minimization of the cardinality of the set of prototypes able to achieve a good clustering quality measure. The balance between these two terms is controlled by $\lambda$ (*penalization coefficient*).

The cooling strategy is implemented in step (6) of Tab. 1. It should be noted that this strategy is only one of the many possible choices. The application of different strategies, and in fact, the value of $\alpha$ itself, can significantly influence the quality of results and the computing time. The decay law and the parameter value used have proven to be reasonable in our experiments, but they may have to be evaluated on the specific application.

## 5 Experimental results

### 5.1 Data set

In order to test our approach, we have used a high-dimensional bioinformatics data set, the publicly available Leukemia data by Golub et al. (1999). The Leukemia problem consists in characterizing two forms of acute leukemia, Acute Lymphoblastic Leukemia (ALL) and Acute Myeloid Leukemia (AML). The original work proposed both a supervised classification task ("class prediction") and an unsupervised characterization task ("class discovery"). Here we obviously focus on the latter, but we exploit the diagnostic information on the type of leukemia to assess the goodness of the clustering obtained.

The data set contains 38 samples for which the expression level of 7129 genes has been measured with the DNA microarray technique (the interesting human genes are 6817, and the other are controls required by the technique). These expression levels have been scaled by a factor of 100. Of these samples, 27 are cases of ALL and 11 are cases of AML. Moreover, it is known that the ALL class is in fact composed by two different diseases, since they are originated from different cell lineages (either T-lineage or B-lineage).

### 5.2 Performance Comparison

We have compared the following approaches:

(1) FCM on the raw data set (RD);
(2) FCM in the Distance Embedding Space (DES) with different prototypes/data ratios;
(3) FCM in the Membership Embedding Space (MES) with different prototypes/data ratios.

Each experiment corresponds to 1000 independent trials, each of them using a different random initialization of memberships in the FCM algorithm. In all trials, the number of clusters was set to 3, and the fuzziness parameter $m$ of FCM was set to 2.

Fig. 1 shows the representation error versus the prototypes/data ratio averaged over 1000 independent trials.

The first approach (standard FCM on original data) obtains a representation error of 17.2%.

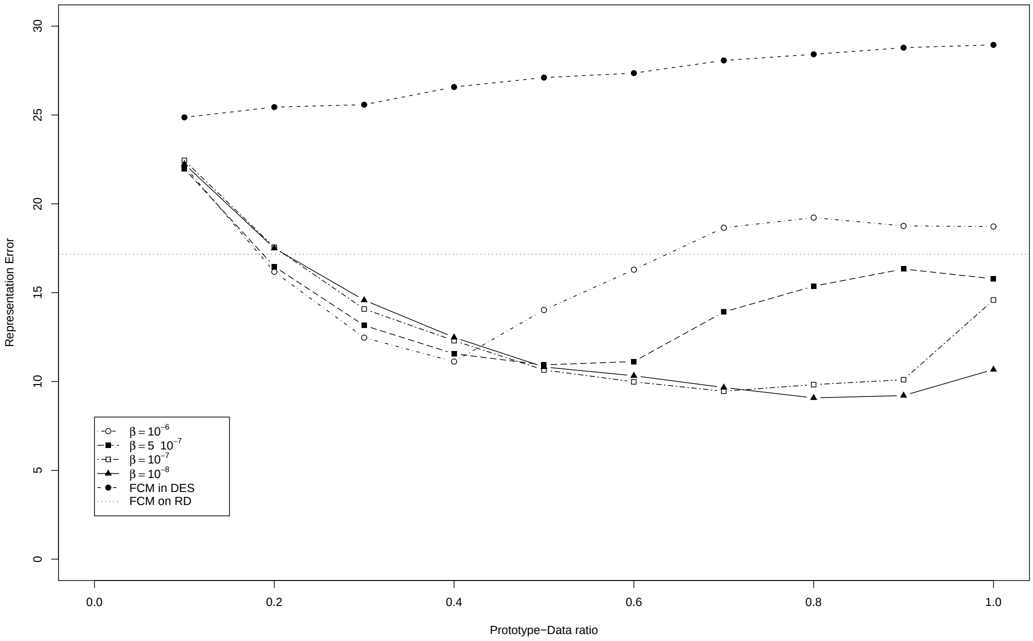

Fig. 1. Representation Error for the tested methods: FCM on row data (RD), FCM on the Distance Embedding Space (DES), FCM on the Membership Embedding Space (MES) with using $\beta = 10^{-6}, 5 \cdot 10^{-7}, 10^{-7}$, and $10^{-8}$.

Table 2
Comparison of the best representation error for the tested methods: FCM on row data (RD), FCM on the Distance Embedding Space (DES), FCM on the Membership Embedding Space (MES).

| Method | $\beta$ | Representation Error | prototypes/data ratio |
|---|---|---|---|
| RD | - | 17.2 | / |
| DES | - | 24.9 | 0.1 |
| MES | $10^{-6}$ | 11.1 | 0.4 |
| MES | $5 \cdot 10^{-7}$ | 10.9 | 0.5 |
| MES | $10^{-7}$ | 9.5 | 0.7 |
| MES | $10^{-8}$ | 9.1 | 0.8 |

The projection onto the distance embedding space (second approach) leads to worse results compared to the first approach: as we can see for Fig. 1, in this case, the representation error is greater than 25.0% for all prototypes/data ratios in the range $[.1, 1.0]$.

The last approach, projecting the data set onto the membership embedding space (MES), leads to better results. In Fig. 1 we show the results with $\beta$ starting from $10^{-6}$ (that is the about the reciprocal of the mean distance between data points) and

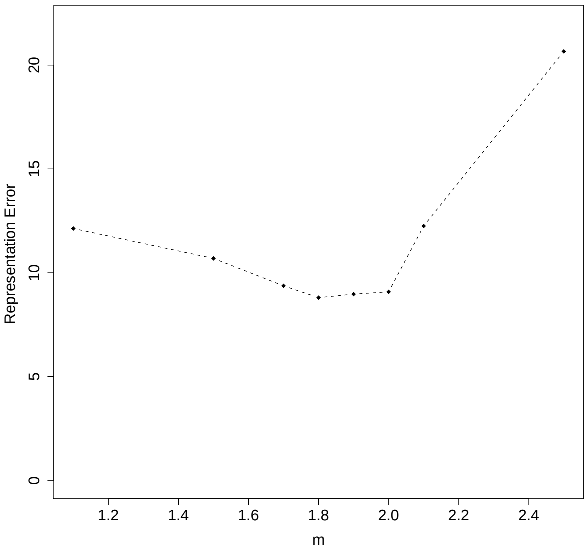

Fig. 2. The behavior of the best error rate (achieved with $\beta = 10^{-8}$, prototypes/data ratio = 0.8) vs the fuzziness parameter *m*.

with decreasing values of this parameter until $10^{-8}$ that gives the optimal representation error.

A comparison of the best representation error for the tested methods is reported in Tab. 2. For each value of $\beta$ we can notice an optimal prototype/data ratio.

Finally we performed a "model selection" to find an appropriate value of the fuzziness *m*, by computing the representation error over 1000 trials. This set of experiments gives also indications about the role of the fuzziness parameter of Fuzzy C-Means when applied in a MES. We obtained the MES using using $\beta = 10^{-8}$ and prototypes/data ratio = 0.8. As shown in Fig. 2, the best value for *m* is $m = 1.8$ that allows to obtain a representation error equal to 8.8% (even if this is slight better than the results obtained by $m = 2$). For $m > 2$ we notice a rapid increasing in the representation error. On the other hand, for low values of *m*, FCM tends to behave like the K-Means algorithm that performs worse than FCM.

### *5.3 Experiments on the Constructive Approach*

We show here the application of the Simulated Annealing Prototype Selection (SA-PS) algorithm to the Leukemia data by Golub et al. (1999) simulating a semi-supervised clustering setting.

We ran the SA-PS algorithm in the MES using the FCM [Bezdek (1981)] algorithm to cluster data. As a clustering quality measure we used the *Representation Error* (RE) evaluated as the best value obtained on $r = 10$ independent trials of FCM.

Table 3
SA-SP algorithm - Choice of parameters.

| *Meaning* | *Symbol* | *Value* |
|---|---|---|
| Number of random perturbations of **g** used to estimate the initial value of $T$ | p | 10000 |
| Number of prototypes to be initially selected | $s_0$ | 3 |
| Cooling parameter | $\alpha$ | 0.9 |
| Membership width parameter | $\beta$ | $10^{-6}$ |
| Maximum number of iteration at each T | $f_{\max}$ | 2000 |
| Minimum number of successes for each T | $h_{\min}$ | 200 |
| Penalization coefficient | $\lambda$ | $10^{-2}$ |
| Minimum number of bits to be switched | $w_{\min}, v_{\min}$ | 1,1 |
| Maximum number of bits to be switched | $w_{\max}, v_{\max}$ | $s$, 5 |
| Number of clusters | $c$ | 3 |
| FCM fuzziness parameter | $m$ | 2 |
| FCM trials | $r$ | 10 |

The parameter $\lambda$ controls the tradeoff between the RE and the number of selected probes (that is a measure of complexity). In our case, the penalization score for each probe corresponds to an RE of 1% ($\lambda = 10^{-2}$). The parameters controlling the annealing are $\alpha$, $f_{\max}$, and $h_{\min}$; we selected $\alpha = 0.9$ to allow a slow cooling of the system and $f_{\max} = 2000$ and $h_{\min} = 200$ in order to have the chance to explore a lot of states for a specific value of $T$. The number of bits to be switched in each move ($w_{\min}$, $v_{\min}$, $w_{\max}$, $v_{\max}$) where selected in order to give the system enough variability to perform small as well as long jumps between states.

Tab. 3 shows the list of parameters of our algorithm and the values we have used in the experiments here reported.

Each independent run of the SA-PS algorithm finds a different small subset of prototypes leading to a clustering Representation Error equal to zero. In Fig. 3, the Representation Error and the number of selected bits of **g** are plotted versus the iteration number during a run of the SA-PS algorithm, where each iteration corresponds to a different value of temperature $T$. In this case, at iterations 31, 33, 34 and 35 we obtained 4 different sets of 3 prototypes giving clustering RE equal to

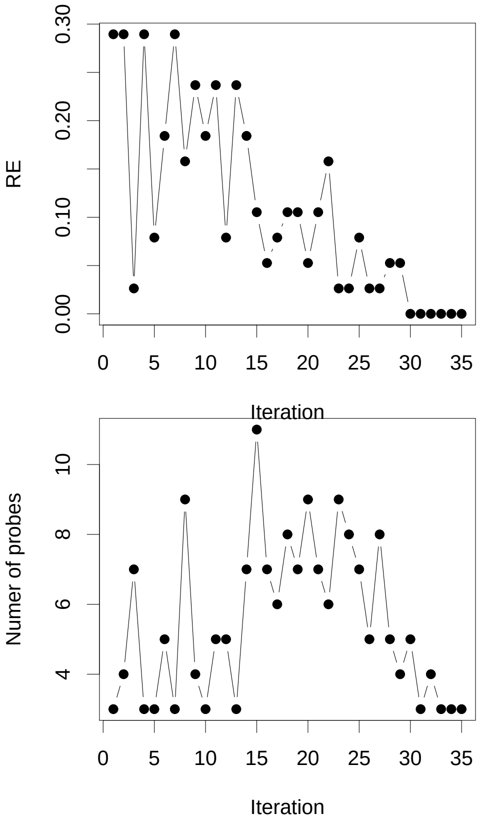

Fig. 3. Representation Error *RE* and number of prototypes selected during a run of the SA-PS algorithm.

zero.

## 6 Conclusions

Clustering methods can achieve poor results when applied to small cardinality and high dimensionality data sets.

In this paper, we proposed a method to face those clustering problems using an embedding space where each data point is represented by a vector containing memberships to fuzzy sets centered on a sub-set of prototypes selected from the data base. On the Leukemia data by Golub et al. (1999) the proposed approach leads to significant improvements with respect the application of clustering algorithms in the original space and in the distance embedding space.

The method can exploit supervised information (class labels) even when these are not available for all data points. This is because they are not used in the optimization step, but only in the centroid evaluation step, which is configured as a model selection over centroid position (a “fitting” criterion) and number (a “complexity” criterion). This makes the proposed approach a viable solution in all cases where supervised information is available, even if only for a subset of data points.

Obtaining (good quality) supervised information has always been an expensive step in setting up an application, but recently this has become an even more serious issue, given the enormous quantities of data that can be produced at a fast pace by sources such as, for instance, enterprise data warehouses, the web, or high-throughput biomolecular analysis techniques. Being able to exploit unsupervised data is important, but perhaps even more important is to be able to exploit even incomplete –but precious– supervised information.

## Acknowledgements

Work funded by the MIUR grant code 2004062740.

## References

Aggarwal, C. C. & Yu, P. S. (2002), ‘Redefining clustering for high-dimensional applications’, *IEEE Transactions on Knowledge and Data Engineering* **14**(2), 210–225.

Bellman, R. (1961), *Adaptive Control Processes: A Guided Tour*, Princeton University Press.

Beyer, K. S., Goldstein, J., Ramakrishnan, R. & Shaft, U. (1999), When is ”nearest neighbor” meaningful?, *in* C. Beeri & P. Buneman, eds, ‘ICDT’, Vol. 1540 of *Lecture Notes in Computer Science*, Springer, pp. 217–235.

Bezdek, J. C. (1981), *Pattern Recognition with Fuzzy Objective Function Algorithms*, Kluwer Academic Publishers, Norwell, MA, USA.

Duda, R. O. & Hart, P. E. (1973), *Pattern Classification and Scene Analysis*, John Wiley and Sons.

Dunn, J. C. (1973), ‘A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters’, *Journal of Cybernetics* **3**, 32–57.

Ester, M., Kriegel, H.-P., Sander, J. & Xu, X. (1996), A density-based algorithm for discovering clusters in large spatial databases with noise, *in* ‘KDD’, pp. 226–231.

Filippone, M. (2009), ‘Dealing with non-metric dissimilarities in fuzzy central clustering algorithms’, *International Journal of Approximate Reasoning* **50**(2), 363–384.

Filippone, M., Camastra, F., Masulli, F. & Rovetta, S. (2008), ‘A survey of kernel and spectral methods for clustering’, *Pattern Recognition* **41**(1), 176–190.

Fred, A. L. N. & Leitão, J. M. N. (2003), ‘A new cluster isolation criterion based on dissimilarity increments', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**(8), 944–958.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D. & Lander, E. S. (1999), 'Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring', *Science* **286**(5439), 531–537.

Guha, S., Rastogi, R. & Shim, K. (1998), Cure: An efficient clustering algorithm for large databases, *in* L. M. Haas & A. Tiwary, eds, 'SIGMOD Conference', ACM Press, pp. 73–84.

Kirkpatrick, S., Jr., D. G. & Vecchi, M. P. (1983), 'Optimization by simulated annealing', *Science* **220**(4598), 671–680.

Lloyd, S. P. (1982), 'Least squares quantization in PCM', *IEEE Transactions on Information Theory* **IT-28**(2), 129–137.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. & Teller, E. (1953), 'Equation of state calculations by fast computing machines', *The Journal of Chemical Physics* **21**(6), 1087–1092.

Ng, R. T. & Han, J. (2002), 'Clarans: A method for clustering objects for spatial data mining', *IEEE Transactions on Knowledge and Data Engineering* **14**(5), 1003–1016.

Pekalska, E., Paclík, P. & Duin, R. P. W. (2001), 'A generalized kernel approach to dissimilarity-based classification', *Journal of Machine Learning Research* **2**, 175–211.

Romeo, F. (1986), Probabilistic hill climbing algorithms: Properties and applications, Technical Report UCB/ERL M86/97, EECS Department, University of California, Berkeley.

Rovetta, S. & Masulli, F. (2006), 'Shared farthest neighbor approach to clustering of high dimensionality, low cardinality data', *Pattern Recognition* **39**(12), 2415–2425.

Shawe-Taylor, J. & Cristianini, N. (2004), *Kernel Methods for Pattern Analysis*, Cambridge University Press.

Steinhaus, H. (1956), 'Sur la division des corp materiels en parties', *Bull. Acad. Polon. Sci* **1**, 801–804.

Strehl, A. & Ghosh, J. (2003), 'Relationship-based clustering and visualization for high-dimensional data mining', *INFORMS Journal on Computing* **15**(2), 208–230.

Černý, V. (1985), 'Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm', *Journal of Optimization Theory and Applications* **45**(1), 41–51.

Zadeh, L. A. (1965), 'Fuzzy sets', *Information and Control* **8**, 338–353.

Zhang, T., Ramakrishnan, R. & Livny, M. (1996), Birch: An efficient data clustering method for very large databases, *in* H. V. Jagadish & I. S. Mumick, eds, 'SIGMOD Conference', ACM Press, pp. 103–114.