A Novel Kernel Method for Clustering

Francesco Camastra, *Member*, *IEEE*, and Alessandro Verri

Abstract—Kernel Methods are algorithms that, by replacing the inner product with an appropriate positive definite function, implicitly perform a nonlinear mapping of the input data into a high-dimensional feature space. In this paper, we present a kernel method for clustering inspired by the classical K-Means algorithm in which each cluster is iteratively refined using a *one-class* Support Vector Machine. Our method, which can be easily implemented, compares favorably with respect to popular clustering algorithms, like K-Means, Neural Gas, and Self-Organizing Maps, on a synthetic data set and three UCI real data benchmarks (*IRIS data*, *Wisconsin breast cancer database*, *Spam database*).

Index Terms—Kernel methods, one class SVM, clustering algorithms, EM algorithm, K-Means.

1 Introduction

Kernel Methods [6], [21] are algorithms that, by replacing the inner product with an appropriate positive definite function [4], implicitly perform a nonlinear mapping of the input data to a high dimensional feature space. While powerful kernel methods have been proposed for supervised classification and regression problems, the development of effective kernel method for clustering, aside for a few tentative solutions [9], [18], is still an open problem.

Tax and Duin [20] and Schölkopf et al. [19] proposed a kernel method, also known as one-class Support Vector Machine (SVM), to characterize the support of a high dimensional distribution. Intuitively, one-class SVM computes the smallest sphere in feature space enclosing the image of the input data. In this paper, we present a kernel method for clustering based on this idea. We start off by initializing K centers in feature space and, for each center, computing the smallest sphere enclosing the *closest* data. Following a K-Means-like strategy, the centers are moved repeatedly by computing, at each iteration and for each center, the smallest sphere enclosing the *closest* data until no center changes anymore. Unlike other popular clustering algorithms [11], our algorithm obtains naturally nonlinear separation surfaces of the data. The plan of the paper is as follows: We recall the main facts of the two methods at the basis of our algorithm, K-Means and one-class SVM, in Sections 2 and 3, respectively. In Section 4, we present and discuss our method. Experiments are reported in Section 5, while we draw our conclusions in Section 6.

2 K-Means

K-Means [14], which we now briefly review, is a classical algorithm for clustering. We first fix the notation: Let $D = \{x_i\}_{i=1}^{\ell}$ be a data set with $x_i \in \mathbb{R}^N$. We call *codebook* the set $W = \{w_k\}_{k=1}^{K}$ with $w_k \in \mathbb{R}^N$ and $K \ll \ell$. The *Voronoi Region* (R_k) of the codevector w_k is the set of all vectors in $\mathbb{R}^N$ for which w_k is the *nearest codevector*

$$R_k = \{x \in \mathbb{R}^N | k = arg \min_{j=1,\ldots,K} \|x - w_j\|\}.$$

• *The authors are with INFM—DISI, Università di Genova, Via Dodecaneso 35, 16146 Genova, Italy. E-mail: camastra@ieee.org, verri@disi.unige.it.*

Manuscript received 19 Apr. 2004; revised 2 Nov. 2004; accepted 8 Nov. 2004; published online 11 Mar. 2005.
Recommended for acceptance by M. Pietikainen.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0184-0404.

The partition of $\mathbb{R}^N$ formed by all Voronoi regions associated with the codebook W is called *Voronoi Tessellation*. The *Voronoi Set* (V_k) of the codevector w_k is the set of all vectors in D for which w_k is the *nearest vector*, that is,

$$V_k = \{x_i \in D \mid k = arg \min_{j=1,\ldots,K} \|x_i - w_j\|\}.$$

For a fixed training set D, the quantization error $E_D(W)$ associated with the Voronoi tessellation induced by the codebook W can be written as

$$E_D(W) = \frac{1}{2\ell} \sum_{k=1}^{K} \sum_{x_i \in V_k} \|x_i - w_k\|^2. \quad (1)$$

K-Means is an iterative method for minimizing the quantization error $E_D(W)$ by repeatedly moving all codevectors to the arithmetic mean of their Voronoi sets. It can be proved [10] that a *necessary* condition for a codebook W to minimize the quantization error in (1) is that each codevector w_k fulfills the *centroid condition*. In the case of finite data set D and Euclidean distance, the centroid condition reduces to

$$w_k = \frac{1}{|V_k|} \sum_{x_i \in V_k} x_i, \quad (2)$$

where $|V_k|$ denotes the cardinality of V_k. Therefore, K-Means is guaranteed to find a local minimum for the quantization error.

The K-Means algorithm consists of the following steps:

1. Initialize the codebook W with K codevectors w_k, $k = 1, \ldots, K$ (each vector w_k drawn randomly without replacement from the set D).
2. Compute the Voronoi Set V_k of each codevector w_k.
3. Move each codevector w_k to the mean of V_k as in (2).
4. Return the codebook if no codevector changed in Step 3, otherwise go to Step 2.

K-Means is an example of an *Expectation-Maximization (EM)* algorithm [5], [7]. The Expectation and Maximization steps are the second and third step, respectively. Since each EM algorithm is always convergent to a local minimum [23], the convergence of K-Means is guaranteed. We note that K-Means is a *batch* algorithm, that is, all inputs are evaluated before any adaptations, unlike *on-line* algorithms, in which the codebook is updated after the evaluation of each input. The main drawback of K-Means is lack of robustness with respect to outliers; this problem can be easily appreciated by looking at the effect of outliers in the computation of the mean in (2).

3 One-Class SVM

We start by recalling the definition of a positive definite kernel, which is at the basis of one-class SVM and kernel methods in general.

Definition 1. *Let X be a nonempty set. A function $G : X \times X \to \mathbb{R}$ is called a* positive definite function *if and only if G is* symmetric *(i.e., $G(x, y) = G(y, x) \ \forall x, y \in X$) and*

$$\sum_{j,k=1}^{n} c_j c_k G(x_j, x_k) \geq 0$$

for all $n \geq 2$, $x_1, \ldots, x_n \subseteq X$ and $c_1, \ldots, c_n \subseteq \mathbb{R}$.

It can be shown [1] that a positive definite function G implicitly defines a mapping $\Phi : X \to \mathcal{F}$ from the input space X to the *feature space* $\mathcal{F}$ endowed with an inner product defined as $G(x, y) = \langle \Phi(x), \Phi(y) \rangle$. Examples of positive definite functions are the *polynomial* kernel

$$G(x,y) = (x \cdot y + 1)^n$$

with x and $y \in \mathbb{R}^N$ and $x \cdot y$ denotes the ordinary inner product between x and y, and the *Gaussian* kernel

$$G(x,y) = e^{-\frac{\|x-y\|^2}{\sigma^2}}.$$

where $\sigma \in \mathbb{R}$.

One-Class SVM [3], [19], [20] is a kernel method based on a support vector description of a data set consisting of positive examples only. If all data are in-liers, one-class SVM computes the smallest sphere in feature space enclosing the image of the input data.

As before, we let $D = \{x_i\}_{i=1}^{\ell}$ with $x_i \in \mathbb{R}^N$ for all i. We start considering the case of one-class SVM in input space. We look for the smallest sphere of radius R that encloses the data x_i. This is described by the constraints:

$$\|x_i - a\|^2 \leq R^2 \quad i = 1 \ldots \ell,$$

where $\|\cdot\|$ is the Euclidean norm and a is the center of the sphere. The constraints can be relaxed by using *slack* variables ξ_i:

$$\|x_i - a\|^2 \leq R^2 + \xi_i \tag{3}$$

with $\xi_i \geq 0$ for all $i = 1, \ldots, \ell$. We solve the problem of finding the smallest sphere introducing the *Lagrangian* $\mathcal{L}$:

$$\begin{aligned}\mathcal{L}(R, a, \xi_1, \ldots, \xi_\ell) = R^2 - \sum_{i=1}^{\ell}(R^2 + \xi_i - \|x_i - a\|^2)\alpha_i \\ - \sum_{i=1}^{\ell}\xi_i\beta_i + C\sum_{i=1}^{\ell}\xi_i,\end{aligned} \tag{4}$$

where $\alpha_i \geq 0$ and $\beta_i \geq 0$ for $i = 1, \ldots, \ell$ are the Lagrange multipliers associated to (3) and to the slack variables, C is a trade-off parameter, and

$$\sum_{j}^{l}\xi_i$$

is the penalty term accounting for the presence of outliers. From the conditions

$$\frac{\partial \mathcal{L}}{\partial R} = \frac{\partial \mathcal{L}}{\partial a} = \frac{\partial \mathcal{L}}{\partial \xi_i} = 0,$$

we get

$$\sum_{i=1}^{\ell}\alpha_i = 1 \tag{5}$$

$$a = \sum_{i=1}^{\ell}\alpha_i x_i \tag{6}$$

$$\alpha_i = C - \beta_i \quad i = 1, \ldots \ell. \tag{7}$$

Using (5), (6), and (7), the constrained minimization of the Lagrangian in (4) can be rewritten as the maximization of the *Wolfe dual form* [2] $\mathcal{L}$:

$$\mathcal{L} = \sum_{i=1}^{\ell}\alpha_i x_i \cdot x_i - \sum_{i=1}^{\ell}\sum_{j=1}^{\ell}\alpha_i\alpha_j x_i \cdot x_j, \tag{8}$$

subject to the constraints

$$0 \leq \alpha_i \leq C \quad i = 1, \ldots, \ell \quad \textit{and} \quad \sum_{i=1}^{\ell}\alpha_i = 1.$$

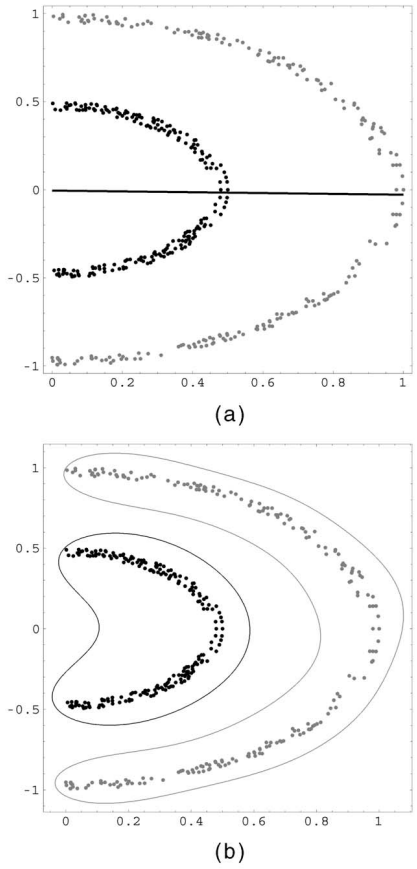

Fig. 1. Comparison between (a) K-Means and (b) our method on the Delta Set. (a) The horizontal line indicates the separation determined by K-Means. (b) The region delimited by the darker and lighter close curves identify the input data, the distance of which in feature space from the two center is less than 0.75 and 0.84, respectively. For our method, we used the Gaussian kernel with $\sigma = 0.4$.

Finally, the Karush-Kuhn-Tucker conditions for $i = 1, \ldots, \ell$ yield

$$\xi_i\beta_i = 0 \tag{9}$$

$$(R^2 + \xi_i - \|x_i - a\|^2)\alpha_i = 0. \tag{10}$$

From (6), it is clear that only the points x_i for which $\alpha_i \neq 0$ are needed for defining the center of the sphere. These points are called Support Vectors. From (9), it follows that, for the points x_i lying outside the sphere, since $\xi_i > 0$, we have $\beta_i = 0$ and, thus, from (7), $\alpha_i = C$. These support vectors are sometimes called bounded support vector (BSV). If $0 < \alpha_i < C$, from (7) and (9) we have that $\beta_i \neq 0$ and, hence, $\xi_i = 0$; therefore, the corresponding support vectors x_i, from (10), lie on the surface of the sphere. Always, from (10), it follows that, for all the points x_i lying inside the sphere, we necessarily have $\alpha_i = 0$.

From the dual form (8), it is clear that all that has been said remains true if the ordinary inner product between input points is replaced by a positive definite function G. Intuitively, this is equivalent to thinking in terms of a one-class SVM working in the feature space induced by the choice of the kernel function G. The dual Lagrangian $\mathcal{L}$ thus becomes

$$\mathcal{L} = \sum_{i=1}^{\ell}\alpha_i G(x_i, x_i) - \sum_{i=1}^{\ell}\sum_{j=1}^{\ell}\alpha_i\alpha_j G(x_i, x_j)$$

with the α_i subject to the same constraints as above.

The only difference with respect to the case of one-class SVM in input space is that, if the mapping Φ is unknown (like for the

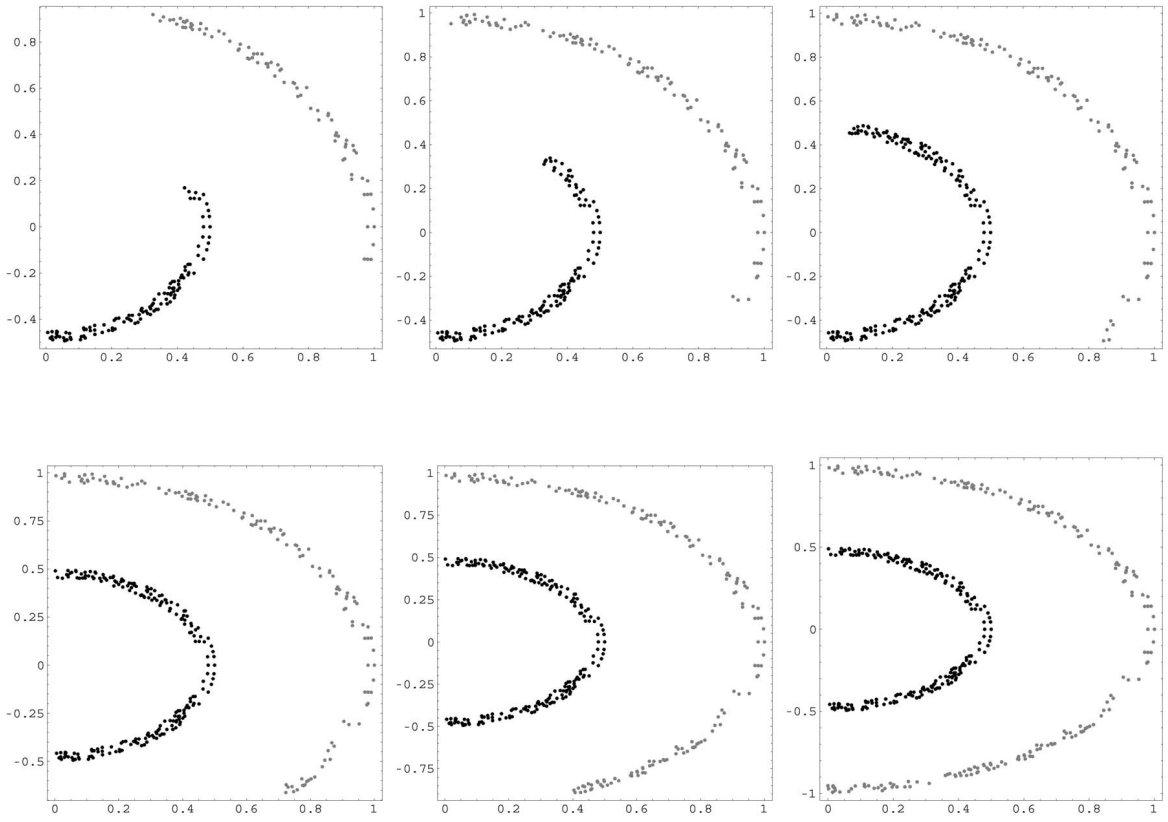

Fig. 2. The six iterations required for convergence of our kernel method on the Delta Set (from upper left to lower right). Black and gray points indicate the data of the first and second cluster, respectively. The points that are not attributed to either one of the clusters in each iteration are not shown. After the last iteration, all the training points ended up in either one of the two clusters.

Gaussian kernel), the center of the sphere in feature space, $\mathcal{A}$, cannot be written explicitly as a linear combination of the image of the training points. However, the distance $R(x)$ in feature space between the image of *any* input point x, $\Phi(x)$, and $\mathcal{A}$ can be evaluated in terms of the kernel function G and the training points x_i. Since $R^2(x) = \Phi(x) \cdot \Phi(x) - 2\Phi(x) \cdot \mathcal{A} + \mathcal{A} \cdot \mathcal{A}$, we have

$$R^2(x) = G(x,x) - 2\sum_{i=1}^{\ell} \alpha_i G(x_i,x) + \sum_{i=1}^{\ell}\sum_{j=1}^{\ell} \alpha_i \alpha_j G(x_i,x_j). \quad (11)$$

Notice that the radius of the sphere in feature space is equal to $R(x_j)$ with x_j any support vector for which $\alpha_j < C$.

4 The Proposed Algorithm

In this section, we describe the kernel method we proposed for clustering. For ease of convenience, we describe our method in some feature space $\mathcal{F}$ assuming we have explicit knowledge of the corrresponding map Φ for which $\xi = \Phi(x)$. The idea is to consider K centers in feature space $\{\mathcal{A}_k\}_{k=1}^{K} \in \mathcal{F}$. We call the set $\mathcal{W} = \{\mathcal{A}_1, \ldots, \mathcal{A}_K\}$ *feature space codebook* since, in our method, the centers in feature space play the same role as the codevectors in input space. In strict analogy with the codevectors in input space, we define the *Voronoi Region* and *Voronoi Set* in feature space for each center $\mathcal{A}_k$.

The *Voronoi Region in feature space*, $\mathcal{R}_k$, of the center $\mathcal{A}_k$ is the set of all feature vectors ξ such that $\mathcal{A}_k$ is the closest vector, or

$$\mathcal{R}_k = \{\xi \in \mathcal{F} \mid k = arg \min_{j=1,\ldots,K} \|\xi - \mathcal{A}_j\|\}.$$

The *Voronoi set in feature space*, $\mathcal{V}_k$, of the center $\mathcal{A}_k$ is the set of all feature vectors $\xi_i = \Phi(x_i)$ in $\mathcal{F}$ such that $\mathcal{A}_k$ is the *closest vector*, or

$$\mathcal{V}_k = \{\xi_i \in \mathcal{F} \mid k = arg \min_{j=1,\ldots,K} \|\xi_i - \mathcal{A}_j\|\}. \quad (12)$$

These definitions induce a *Voronoi tessellation of the feature space*. Attracted by the idea of support vector description of data sets, the key point of the proposed method is to describe each cluster by a

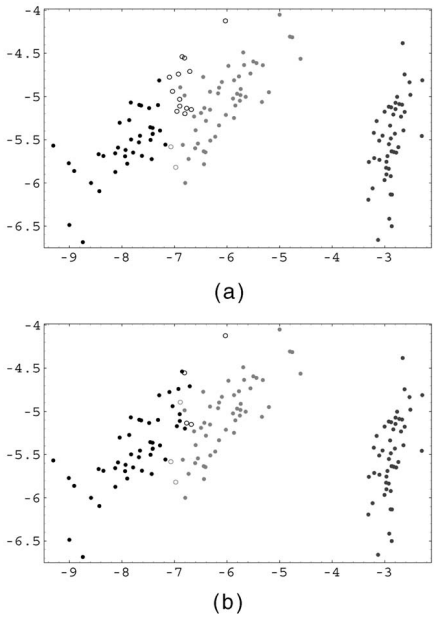

Fig. 3. (a) K-Means and (b) our Kernel Method on the IRIS Data Set. Each data class is represented with a different gray level. Filled disks and circles indicate correctly classified and misclassified data, respectively. For our method, we use a Gaussian kernel with $\sigma = 1.1$.

TABLE 1
SOM, K-Means, Neural Gas, Ng-Jordan Algorithm, and Our Kernel Method Average Performances in Terms of Correctly Classified Points on IRIS, Wisconsin, and Spam Database

Algorithm	Iris Data	Wisconsin Database	Spam Database
SOM	121.5 ± 1.5 (81.0%)	660.5 ± 0.5 (96.7%)	1210 ± 30 (78.9%)
K-Means	133.5 ± 0.5 (89.0%)	656.5 ± 0.5 (96.1%)	1083 ± 153 (70.6%)
Neural Gas	137.5 ± 1.5 (91.7%)	656.5 ± 0.5 (96.1%)	1050 ± 120 (68.4%)
Ng-Jordan Algorithm	126.5 ± 7.5 (84.3%)	652 ± 2 (95.5%)	929 ± 0 (60.6%)
Our kernel method	142 ± 1 (94.7%)	662.5 ± 0.5 (97.0%)	1247 ± 3 (81.3%)

For our method, we used a Gaussian Kernel with $\sigma = 1.1$ for the IRIS Data, $\sigma = 0.9$ for the Wisconsin Data, and $\sigma = 2.0$ for the Spam Database.

sphere of minimum radius. The assignment of which point to which cluster and the spheres of minimum radius are obtained through an iterative procedure similar to K-Means. Starting off with a cluster initialization based on a small number of points, at each iteration, a one-class SVM is trained on each cluster and the obtained spheres are used to compute the Voronoi set $\mathcal{V}_k$. The procedure stops when no Voronoi set changes. In the attempt to gain robustness against outliers, an interesting variant (which in principle could also applied to K-Means) is to modify the notion of Voronoi set by defining $\mathcal{V}_k(\rho)$ as

$$V_k(\rho) = \{\xi_i \in \mathcal{F} \mid k = arg \min_{j=1,\ldots,K} \|\xi_i - \mathcal{A}_j\| \ \ and \ \ \|\xi_i - \mathcal{A}_k\| < \rho\}. \tag{13}$$

In words, the Voronoi set of the center $\mathcal{A}_k$ *includes only* the data points in which the distance in feature space is *smaller* than ρ. The parameter ρ can be chosen, for example, using model selection techniques [5]. As we have seen in the previous section, the fact that the centers of the sphere of smallest radius cannot be explicitly expressed if the kernel mapping is only implicitly defined does not undermine the method.

For a fixed choice of a kernel function G, our algorithm consists of the following steps:

1. Initialize K Voronoi sets $\mathcal{V}_k(\rho)$, $k = 1, \ldots, K$ using a small subset of the ℓ training points.
2. Train a one-class SVM for each $\mathcal{V}_k(\rho)$.
3. Update each $\mathcal{V}_k(\rho)$.
4. If no Voronoi set changes in Step 3, exit; otherwise, go to Step 2.

The convergence of this procedure is an open problem. In practice, with $C = 1$ (that is, all points in each Voronoi set $\mathcal{V}_k(\rho)$ are contained within the sphere of smallest radius) and ρ constant across iterations, the procedure always converged in all the performed experiments (over one thousand runs). The intuition is that, under these conditions, Steps 2 and 3 mimic an Expectation and a Maximization steps. The similarity with K-Means is clear but we notice three important differences. First, the algorithm does not aim at minimizing the quantization error because the Voronoi sets are not based on the computation of the centroid (not even in the feature space). Second, depending on the choice of the constant ρ, not all points are necessarily assigned to one of the K clusters. Third, since the expression of the centers of the spheres in feature space might not be available, the codevectors might be defined only implicitly.

5 Experimental Results

Our algorithm has been tried on a synthetic data set (*Delta Set*) and on three UCI data sets, that is, the *IRIS* Data, the *Wisconsin's breast cancer* database, and the *Spam* data set.

Delta Set[1] is a 2D set formed by 424 points of two linearly nonseparable classes. Therefore, the two classes cannot be separated by K-Means using only two codevectors (see Fig. 1a). K-Means shares this limitation with other clustering algorithms, like SOM [12], [13] and Neural Gas [15]. We then applied our kernel method to the Delta Set using only two centers. As shown in Fig. 1b, our algorithm can separate the two clusters. It is important to remark that the counterimages of the centers in input space do not exist. Fig. 2 shows the algorithm behavior over the six stages required for convergence. IRIS Data,[2] possibly the most used real data benchmark in Machine Learning proposed by Fisher [8] in 1936, is formed by 150 points belonging to three different classes. One class is linearly separable from the other two, while the other two are not. IRIS Data is usually represented, projecting the original 4D data along the two major principal components. We tested K-Means, Neural Gas, SOM, and our method on the IRIS data using one center for each of the three classes. We also compared the obtained results against the *Ng-Jordan algorithm* [17], a spectral clustering algorithm [16]. The results obtained using K-Means and our algorithm is shown in Fig. 3. The second column of Table 1 shows the average performances on 20 runs of K-Means, Neural Gas, SOM, Ng-Jordan algorithm, and our method obtained with different initializations and parameters. From the displayed figures, it can be seen that our algorithm appears to perform better than the other algorithms.

The Wisconsin's breast cancer database,[3] proposed by Wolberg and Mangasarian [22] in 1990, collects 699 cases of diagnostic samples. After the removal of the 16 database samples with missing values, the database consists of 683 9D feature vectors belonging to two different classes, benign and malignant tumors. The Spam database[4] collects 1,534 samples from two different classes, spam and not-spam. Each sample is represented by a 57-dimensional feature vector. Here, again, we tested K-Means, Neural Gas, SOM, Ng-Jordan algorithm, and our method on both the Wisconsin and Spam databases using one center for either of the two classes. The third and fourth column of Table 1 display the average performances on 20 runs obtained on Wisconsin and Spam databases, respectively, for different initializations and parameters. As before, it can be seen that our method obtains consistently better results than the other algorithms.

1. The Delta data set can be downloaded from the following ftp address: ftp.disi.unige.it/person/CamastraF/delta.dat.
2. The IRIS Data can be downloaded from the following ftp address: ftp.ics.uci.edu/pub/machine-learning-databases/iris.
3. Wisconsin's breast cancer database can be downloaded from the following ftp address: ftp.ics.uci.edu/pub/machine-learning-databases/breast-cancer-wisconsin.
4. The spam database can be downloaded from the following ftp address: ftp.ics.uci.edu/pub/machine-learning-databases/spam.

6 Conclusion

In this paper, we have proposed a batch clustering method inspired by K-Means and based on one-class SVM description of a data set. Our method, whose empirical convergence has been consistently verified, compares favorably against popular clustering algorithms, like K-Means, Neural Gas, and Self-Organizing Maps, on a synthetic data set and three UCI benchmarks, *IRIS data*, *Wisconsin breast cancer database*, and *Spam database*. Future work includes the study of the theoretical properties of the method (in particular, convergence) and extension of the experimental validation to computer vision applications, like image and video segmentation.

Acknowledgments

The authors are indebted to the anonymous reviewers for their valuable comments. The authors would like to thank A.Vinciarelli for useful discussions. This research has been partially funded by the FIRB Project RBAU01877P (ASTA), INFM PRA project, MAIA, and the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778.

References

[1] N. Aronszajn, "Theory of Reproducing Kernels," *Trans. Am. Math. Soc.*, vol. 686, pp. 337-404, 1950.
[2] M. Bazaraa and C.M. Shetty, *Nonlinear Programming*. New York: Wiley, 1979.
[3] A. Ben-Hur, D. Horn, H.T. Siegelmann, and V. Vapnik, "Support Vector Clustering," *J. Machine Learning Research*, vol. 2, pp. 125-137, 2001.
[4] C. Berg, J.P.R. Christensen, and P. Ressel, *Harmonic Analysis on Semigroups*. New York: Springer-Verlag, 1984.
[5] C. Bishop, *Neural Networks for Pattern Recognition*. Cambridge: Cambridge Univ. Press, 1995.
[6] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge: Cambridge Univ. Press, 2000.
[7] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistical Soc.*, vol. 39, no. 1, pp. 1-38, 1977.
[8] R.A. Fisher, "The Use of Multiple Measurements in Taxonomic Problems," *Annals of Eugenics*, vol. 7, pp. 179-188, 1936.
[9] M. Girolami, "Mercer Kernel Based Clustering in Feature Space," *IEEE Trans. Neural Networks*, vol. 13, no. 3, pp. 780-784, 2002.
[10] R.M. Gray, *Vector Quantization and Signal Compression*. Dordrecht: Kluwer Academic Press, 1992.
[11] A.K. Jain, M.N. Murty, and P.J. Flynn, "Data Clustering: A Review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264-323, 1999.
[12] T. Kohonen, "Self-Organized Formation of Topologically Correct Feature Maps," *Biological Cybernetics*, vol. 43, no. 1, pp. 59-69, 1982.
[13] T. Kohonen, *Self-Organizing Map*. New York: Springer-Verlag, 1997.
[14] S.P. Lloyd, "An Algorithm for Vector Quantizer Design," *IEEE Trans. Comm.*, vol. 28, no. 1, pp. 84-95, 1982.
[15] T.E. Martinetz and K.J. Schulten, "Neural-Gas Network for Vector Quantization and Its Application to Time-Series Prediction," *IEEE Trans. Neural Networks*, vol. 4, no. 4, pp. 558-569, 1993.
[16] M. Meila, "Comparing Clusterings," Technical Report 418, Dept. of Statistics, Univ. of Washington, 2003.
[17] A.Y. Ng, M.I. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an Algorithm," *Advances in Neural Information Processing Systems 14*, pp. 849-856, 2001.
[18] B. Schölkopf, A.J. Smola, and K.-R Müller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," Technical Report No. 44, Max Planck Institut für Biologische Kybernetik, 1996.
[19] B. Schölkopf, R.C. Williamson, A.J. Smola, J. Shawe-Taylor, and J. Platt, "Support Vector Method for Novelty Detection," *Advances in Neural Information Processing Systems 12*, pp. 526-532, 1999.
[20] D.M.J. Tax and R.P.W. Duin, "Support Vector Domain Description," *Pattern Recognition Letters*, vol. 20, nos. 11-13, pp. 1191-1199, 1999.
[21] V. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.
[22] W.H. Wolberg and O.L. Mangasarian, "Multisurface Method of Pattern Separation for Medical Diagnosis Applied to Breast Cytology," *Proc. Nat'l Academy of Sciences, USA*, vol. 87, pp. 9193-9196, 1990.
[23] C.F.J. Wu, "On the Convergence Properties of the EM Algorithm," *The Annals of Statistics*, vol. 11, no. 1, pp. 95-103, 1983.