

GAUSSIAN KERNEL WIDTH GENERATOR FOR SUPPORT VECTOR CLUSTERING

SEI-HYUNG LEE, KAREN DANIELS

*Department of Computer Science, University of Massachusetts Lowell
One University Avenue, Lowell, MA 01854 USA
phone: +1-978-934-3631, fax: +1-978-934-3551
Email: {slee, kdaniels}@cs.uml.edu*

Clustering data into natural groupings has important applications in fields such as Bioinformatics. Support Vector Clustering (SVC) does not require prior knowledge of a dataset and it can identify irregularly shaped cluster boundaries. A major SVC challenge is the choice of an important parameter value, the width of a kernel function that determines a nonlinear transformation of the input data. Since evaluating the result of a clustering algorithm is a highly subjective process, a collection of different parameter values must typically be examined. However, no algorithm has been proposed to specify the parameter values. This paper presents a secant-like numerical algorithm that generates an increasing sequence of SVC kernel width values. An estimate of sequence length depends on spatial characteristics of the data but not the number of data points or the data's dimensionality. The algorithm relies on a function that relates the kernel width value to the radius of the minimal sphere enclosing the images of data points in a high-dimensional feature space. Experimental results with 2D and higher-dimensional datasets suggest that the algorithm yields useful data clusterings.

Keywords: Cluster analysis, Support vector clustering, Secant method, Lagrange multipliers

1. Introduction

1.1. Clustering Overview

Clustering is a natural grouping or unsupervised classification of the data into groups [7]. Clustering has many important applications in fields such as Bioinformatics. Unfortunately, clustering is a subjective problem, since there is no unique definition of a cluster or unique answer for a given dataset. Because of this, research has produced many different types of clustering criteria and clustering algorithms (see [7,9,20] for clustering surveys and

reviews).

Most clustering algorithms use one or a combination of the following techniques: graph-based [6,18,19,8], density-based [17], model-based methods using either a statistical approach or a neural network approach, or optimization of a clustering criterion function. Fast clustering algorithms tend to depend heavily on parameter selection. If no prior information about the dataset is available, argument-free clustering algorithms like the hierarchical approach of [21] can be used.

Finding boundaries of clusters is another popular technique [19,2]. Support Vector Clustering (SVC) is a boundary-finding clustering method that does not require prior knowledge about the data. It is based on concepts from support vector machines. A support vector machine is a classifier that is used widely in many machine learning applications [13].

This paper is based on SVC, which is described below in Section 1.2. A key limitation of SVC is described in Section 1.3. Section 1.4 summarizes this paper's contribution that addresses the limitation and gives an overview of the remainder of the paper.

1.2. *Support Vector Clustering*

As in support vector machines, SVC uses a nonlinear mapping of the data into a high-dimensional feature space. In both techniques the feature space is a Hilbert space for which a kernel function defines an inner product. Whereas support vector machines use a linear separator in the feature space in order to separate and classify points, SVC uses a minimal sphere encompassing feature space images of data points. In SVC, the minimal sphere in feature space can be mapped into contours in the data space. These contours are interpreted as clusters [4]. Cluster boundaries can be approximated using data point images that lie on the surface of the minimal sphere. A significant advantage of SVC over other clustering methods is its ability to produce irregularly shaped clusters. Another advantage is its ability to handle outliers in a dataset. Outliers are accommodated by allowing the minimal sphere to exclude some data point images.

Given a finite set $\mathcal{X} \subseteq \mathcal{R}^d$ of N distinct data points as in [4], the minimal sphere of radius R enclosing all data points' images in the feature space can be described by the following, as in [4]:

$$\|\Phi(x) - a\|^2 \leq R^2 \quad \forall x \in \mathcal{X}, \quad (1)$$

where Φ is a nonlinear mapping from data space to feature space, $\Phi(x)$ is

the feature space image of data point x , $\|\cdot\|$ is the Euclidean norm, and a is the center of the sphere. Images on the surface of the sphere correspond to points on contour boundaries in data space. Images inside the sphere map into points within contours.

The mapping from data space to feature space is governed by a kernel function, $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{R}$, that defines the inner product of image points. Throughout this paper, we assume that the Gaussian kernel given by Eq. (2) below is used as a kernel function:

$$K(x_i, x_j) = e^{-q\|x_i - x_j\|^2} = \langle \Phi(x_i) \cdot \Phi(x_j) \rangle, \quad (2)$$

where q is the width of the Gaussian kernel. From (2), it immediately follows that,

$$K(x_i, x_i) = 1 \quad (3)$$

From Eq. (1) to (3), the Lagrangian W below can be derived, which must be maximized in order to find the minimal sphere:

$$\begin{aligned} \max W &= \sum_j K(x_j, x_j)\beta_j - \sum_{i,j} \beta_i\beta_j K(x_i, x_j) \\ &= 1 - \sum_{i,j} \beta_i\beta_j K(x_i, x_j) \end{aligned} \quad (4)$$

where β_i is a Lagrange multiplier, $1 \leq i \leq N$, and $\sum_i \beta_i = 1$. A soft margin constraint of the form $0 \leq \beta_i \leq C$, where C is a constant, controls the number of Bounded Support Vectors (BSVs), that is, the number of data points that will be considered as outliers. Eq. (4) is quadratic in the β values. Solving Eq. (4) using quadratic programming yields β values corresponding to the minimal sphere in feature space. If $\beta_i = C$, then $\Phi(x_i)$ is outside the minimal sphere and x_i is a BSV. If $0 < \beta_i < C$, then $\Phi(x_i)$ is on the boundary of the minimal sphere and x_i is a Support Vector (SV). Otherwise, $\beta_i = 0$ and $\Phi(x_i)$ is interior to the minimal sphere.

If the images of all points on the line segment connecting two data points has smaller distance from a than R^2 , then the two data points are regarded as being in the same cluster and an edge between them is added to an adjacency matrix. A cluster is a connected component of the adjacency matrix.

1.3. *Support Vector Clustering Limitation*

Of particular interest in this paper is the parameter q that specifies the width of the Gaussian kernel. The width controls how "spread out" the data points' feature space images are and therefore determines the size of the minimal sphere. The number of q values is a multiplicative factor in SVC running time, so finding a small set of q values for a given dataset that reflect all of the natural clusterings of the data is one of the most significant challenges of SVC. In [4], an initial q value producing one cluster is suggested. However, an algorithm is needed for generating additional q values beyond the initial one.

1.4. *Contribution and Overview*

In Section 2, we offer an algorithm for exploring SVC Gaussian kernel width for a fixed value of the outlier parameter^a C . An increasing sequence of kernel width values is generated using properties of the minimal sphere. The secant-like numerical algorithm applies to datasets of arbitrary dimension. An estimate is provided for the number of kernel width values generated by the algorithm. The estimate depends on spatial characteristics of the dataset and is independent of dataset dimension as well as the number of data points. Section 3 gives experimental results of our approach, demonstrating that it often provides useful clusterings in practice. Section 4 concludes the paper and describes future work.

2. Kernel Width Sequence Generator

This section presents our approach for generating an increasing sequence of kernel width values. It is based on the intuition that significant changes in clustering structure are less likely to occur in q intervals where R^2 values are fairly stable. A secant-like, numerical algorithm is proposed in Section 2.2. This relies on R^2 monotonicity. Therefore, this section starts by characterizing R^2 as a function of q . The results of Section 2.1 establish that $0 \leq R^2 \leq 1 - 1/N$ for $0 \leq q \leq \infty$, $R^2 = 0$ for $q = 0$, $R^2 = 1 - 1/N$ if and only if $q = \infty$, and provide conditions under which R^2 is a monotonically nondecreasing function of q . Results of this section hold for arbitrary input space dimension. We assume that the value of C is fixed so that the

^aThis paper is an enhancement of our technical report [1] in which outliers are not allowed.

number of outliers is not varied. We estimate the number of q values generated by the secant-like algorithm using known results on secant method convergence. The estimate relies on spatial characteristics of the dataset but not the number of data points or dimensionality of the dataset.

2.1. Characterizing R^2 as a function of q

All proofs of lemmas and theorems are omitted and can be found in [1].

Lemma 2.1. $\frac{1}{N} < C \leq 1$.

To characterize R^2 as a function of q , we first observe that, as R is the radius of the minimal sphere enclosing data point images, $R^2 \geq 0$ for $0 \leq q \leq \infty$. Furthermore, Lemma 2.2 below shows that if $q = 0$, then $R^2 = 0$.

Lemma 2.2. $q = 0$ if and only if $R^2 = 0$.

Since the kernel is Gaussian, Eq. (3) implies that the entire data space is embedded onto the surface of the unit ball in feature space [13]. Therefore, $R^2 \leq 1$ for $0 \leq q \leq \infty$ and hence $0 \leq R^2 \leq 1$ for $0 \leq q \leq \infty$. Lemmas 2.3-2.5 and Corollary 2.1.1 below together establish the stronger result that $R^2 \leq 1 - 1/N$. The proof of Lemma 2.5 below relies on Lemma 2.4.

Lemma 2.3. If $q = \infty$, then $\beta_i = \frac{1}{N}$ for all $i \in \{1, \dots, N\}$.

Lemma 2.4. If $q = \infty$, then $R^2(x_i) = 1 - \frac{1}{N}$.

Lemma 2.5. $R^2 = 1$ if and only if $q = \infty$ and $N = \infty$.

Corollary 2.1.1. If N is finite, then $R^2 \leq 1 - \frac{1}{N} < 1$. Furthermore, $R^2 = 1 - \frac{1}{N}$ if and only if $q = \infty$.

The results above combined with Lemma 2.6 and Theorem 2.1 below show that, under certain conditions, R^2 is a monotonically nondecreasing function of q . In the following results the q subscript indicates kernel width q .

Lemma 2.6. If $q' > q$,
then $\|\Phi(x_i) - \Phi(x_j)\|_{q'} > \|\Phi(x_i) - \Phi(x_j)\|_q$.

Now, let $S_q(\mathcal{X})$ denote the minimal sphere for point set \mathcal{X} and kernel value q . It follows from Welzl's minimal sphere algorithm of [3] that $S_q(\mathcal{X})$ in an s -dimensional space is determined by a set V of points (support

vectors), such that $S_q(V) = S_q(\mathcal{X})$, whose images lie on the surface of $S_q(\mathcal{X})$, where $2 \leq |V| \leq \min(N, s+1)$. V is *minimal* if there does not exist $V' \subseteq V$ such that $S_q(V') = S_q(V)$.

Theorem 2.1. *For $q' > q$ and $C = 1$ in an s -dimensional feature space, suppose that there exists a set V of minimal support vectors for q' and q such that $S_{q'}(V) = S_{q'}(\mathcal{X})$, $S_q(V) = S_q(\mathcal{X})$, and $|V| \leq 3$. Then $R_{q'}^2 \geq R_q^2$.*

The proof of Theorem 2.1 uses Lemma 2.6 and Corollary 2.1.1.

Theorem 2.2. *For $C = 1$, $\frac{\max_{ij} \{\|\Phi(x_i) - \Phi(x_j)\|_q\}}{2}$ is a monotonically non-decreasing lower bound on R as a function of q .*

Corollary 2.1.2. *If x_i and x_j are the maximally separated pair of points in the data space, then, for $C = 1$, $\frac{1 - K_q(x_i, x_j)}{2}$ is a monotonically non-decreasing lower bound on R^2 as a function of q .*

Theorem 2.3. *For $\frac{1}{N} < C \leq 1$, $\frac{\min_{ij} \{\|\Phi(x_i) - \Phi(x_j)\|_q\}}{2}$ is a monotonically nondecreasing lower bound on R as a function of q .*

Corollary 2.1.3. *If x_i and x_j are the minimally separated pair of points in the data space, then $\frac{1 - K_q(x_i, x_j)}{2}$ is a monotonically nondecreasing lower bound on R^2 as a function of q .*

Note that these lower bounds are $= 0$ for $q = 0$ and $= \frac{1}{2}$ for $q = \infty$.

2.2. Secant Algorithm

The operation of the secant procedure is illustrated in Figure 1. Since Section 2.1 showed that $R^2 \geq 0$ for $q \geq 0$ and $R^2 = 0$ for $q = 0$, the starting q value for the secant procedure is $q = 0$. The second q value is from [4] and is expected to yield a result of one cluster. For each value of q , the associated R^2 value is calculated using the SVC steps of updating a kernel matrix, solving the Lagrangian, and computing the radius of the minimal sphere. To generate each subsequent q value, a line through the two previous R^2 curve points is extended until it intersects the line $R^2 = 1 - 1/N$. The secant algorithm terminates when every data point is an SV or the slope of the line is close to flat. When every data point is an SV, the number of clusters is typically N and no useful clustering information is usually gained for larger q values.

Since the goal of this algorithm is to generate an increasing sequence Q of q values, whenever the new q value q_{new} is less than the previous

one q_{old} , monotonicity of the R^2 curve is enforced using several techniques. The first technique is *support vector permanence*. This requires that each support vector for q_{old} is also a support vector for q_{new} . This is motivated by the conditions of Theorem 2.1. Support vector permanence also requires that no new BSVs be added for q_{new} beyond the BSVs that exist for q_{old} . Support vector permanence can be achieved by adding to the quadratic program for q_{new} : 1) a constraint of the form $\beta_i > 0$ for each x_i that is an SV for q_{old} and 2) a constraint of the form $\beta_i < C$ for each x_i that is not a BSV for q_{old} .

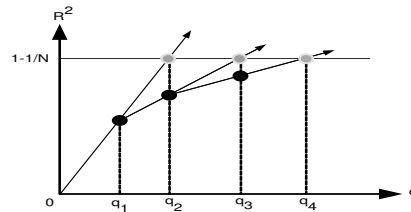


Figure 1. Example: $Q = \langle q_1, q_2, q_3, q_4, \dots \rangle$

Figure 2 is an example of a dataset and the graph of R^2 vs. q generated by the secant method. Multiple curves are shown on the R^2 graph. Each corresponds to a graph of $R^2(x_i)$ for a data point x_i . The R^2 curve is an upper bound on each $R^2(x_i)$ curve.

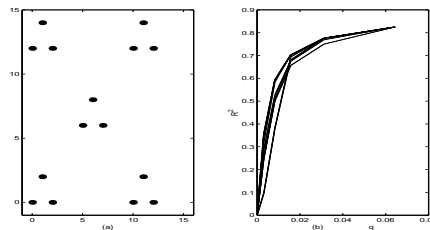


Figure 2. R^2 as a function of q : (a) a dataset ($N=15$); (b) R^2 and $R^2(x_i)$ graphs for $N = 15$ dataset. (q, R^2) values from secant algorithm: $(0.005, 0.47)$, $(0.010, 0.63)$, $(0.019, 0.72)$, $(0.039, 0.79)$, $(0.081, 0.83)$.

The worst-case running time of our algorithm is in $O(|Q|(N^3 + T))$, where $|Q|$ is the number of q values produced and T is the running time of the procedure to enforce monotonicity. To derive an estimate on $|Q|$,

we first describe starting and ending q values q_{min} and q_{max} , respectively, then invoke results on secant method convergence. If all kernel values are equal to 1, then SVC yields 1 cluster; this occurs when $q = 0$. Analogously, if the kernel matrix is the identity matrix, then SVC yields N clusters; this occurs when $q = \infty$. To derive practical values for q_{min} and q_{max} that can still be expected to yield 1 and N clusters, respectively, we force all kernel values to be close to 1 or all off-diagonal kernel values to be close to 0. The former case yields, for $0 \leq \epsilon_{min} \leq 1$:

$$1 - \frac{1}{e^{q(\max\{\|x_i - x_j\|^2\})}} \leq \epsilon_{min}, \Rightarrow q_{min} = \frac{\ln(\frac{1}{1-\epsilon_{min}})}{\max\{\|x_i - x_j\|^2\}}. \quad (5)$$

In the latter case we obtain, for $0 \leq \epsilon_{max} \leq 1$:

$$\frac{1}{e^{q(\min\{\|x_i - x_j\|^2\})}} \leq \epsilon_{max}, \Rightarrow q_{max} = \frac{\ln(\frac{1}{\epsilon_{max}})}{\min\{\|x_i - x_j\|^2\}} \quad (6)$$

$|Q|$ can be estimated using the convergence rate of a secant method, which is in between linear and quadratic [23]. We therefore expect, experimentally, for $|Q|$ to be in between approximately $lg(\frac{q_{max}}{q_{min}})$ and $lg^2(\frac{q_{max}}{q_{min}})$, assuming we start at q_{min} and stop at q_{max} . For linear convergence, we obtain:

$$\begin{aligned} |Q| \approx & lg\left(\frac{\frac{\ln(\frac{1}{\epsilon_{max}})}{\min\{\|x_i - x_j\|^2\}}}{\frac{\ln(\frac{1}{1-\epsilon_{min}})}{\max\{\|x_i - x_j\|^2\}}}\right) = lg(\max\{\|x_i - x_j\|^2\}) \\ & - lg(\min\{\|x_i - x_j\|^2\}) + lg(\ln(\frac{1}{\epsilon_{max}})) \\ & - lg(\ln(\frac{1}{1-\epsilon_{min}})) \end{aligned} \quad (7)$$

For the secant procedure we initialize q_{new} by choosing q_{min} such that $\ln(\frac{1}{1-\epsilon_{min}}) = 1$. Experimentally, this choice is expected to produce 1 cluster [4]. For our analysis we choose q_{max} such that $\ln(\frac{1}{\epsilon_{max}}) = 1$. Experimentally, this can be expected to produce N clusters. Under these assumptions, Eq. 7 simplifies to:

$$|Q| \approx lg(\max\{\|x_i - x_j\|^2\}) - lg(\min\{\|x_i - x_j\|^2\}). \quad (8)$$


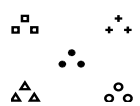
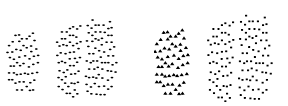
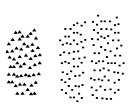


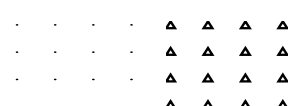





Note that Eq. 8 depends only on spatial characteristics of the dataset and not on the dimensionality of the dataset or the number of data points. Also note that our secant procedure does not explicitly scale the data. The first non-zero q value is a function of the maximum distance between data points. Thus, it compensates for the scale of the data.

3. Results

In this section, we show the results of our algorithm applied to some SVC datasets. Sequential Minimal Optimization (SMO)^b solves the SVC quadratic program efficiently [22]. SVC MATLABTM code was partially used^c.

Table 1 shows clustering results for five 2D datasets. The dataset for $N = 180$ is from [19] and $N = 198$ is from [4]. The $N = 250$ dataset is similar to [4] but without outliers. Table 2 shows the full q sequence produced by our algorithm for each of the five 2D datasets. Kernel width values from Table 1 are checked in Table 2. The R^2 value corresponding to each q is also shown, along with the number of support vectors. The $N = 500$ dataset is from [4] with outliers. The $N = 500$ case is not shown since its result is very similar to the $N = 250$ case.

Table 1. Cluster results. Each box contains a picture of the dataset (left) plus one or more q values with clustering results (right)

 $N = 15$	 $q=0.031$ 5 clusters	 $N = 180$	 $q=0.00049$ 2 cluster	 $N = 250$	 $q=0.00533$ 3 clusters
 $N = 16$	 $q=0.147$ 1 clusters	 $N = 198$	 $q=50$ 4 clusters	 $N = 250$	 $q=324$ 3 clusters

For high-dimensional datasets without outliers that we created, the result supports the observation in ⁴ that increasing numbers of support vectors are needed to represent contours as data dimensionality rises. However, in our tests, SVC is successfully used to obtain correct clusterings in spite of the fact that the number of support vectors is relatively large.

For the iris data of Fisher (1936) without the class label, we clustered it after applying Sammon's nonlinear mapping^[5] and obtained the three

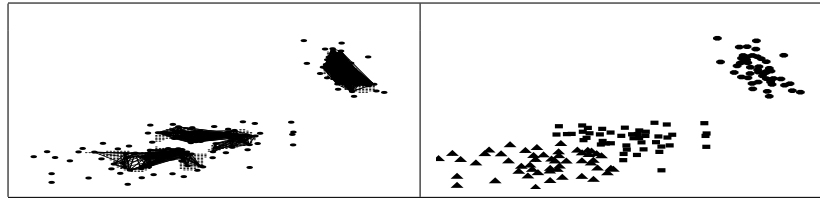
^bSMO code was obtained from OSU SVM(http://www.eleceng.ohio-state.edu/~maj/osu_svm/).
^cSVC MATLABTM code was obtained from (<http://www.cs.tau.ac.il/~borens/courses/ml/code.html>).

Table 2. Sequence of q values. N_{sv} is number of support vectors. N_{cl} is number of clusters.

N	q	N_{sv}	N_{cl}	N	q	N_{sv}	N_{cl}	N	q	N_{sv}	N_{cl}
15	0.003	4	1	✓	15	26	1	500	4036	246	98
	0.008	4	1		27	40	3		5933	249	148
	0.016	6	1		50	58	4		8466	250	187
	0.031	11	5		89	84	4		0.02	3	1
✓	0.064	15	5	160	107	4	0.07	5	1		
	0.056	4	1	278	153	8	0.13	9	1		
	✓	0.147	4	1	463	178	35	0.27	14	2	
	0.273	4	4	729	189	69	0.49	31	1		
16	0.549	16	1	1110	197	98	0.88	46	3		
	180	0.00001	3	1	1656	197	125	1.53	74	2	
	0.00004	6	1	2427	197	152	2.61	103	3		
	0.00016	9	1	3508	197	171	4.35	141	8		
✓	0.00025	14	1	250	2.4	5	1	7.27	182	20	
	0.00049	19	2	7	7	1	12	272	34		
	0.00093	30	2	15	9	1	19	339	69		
	0.00171	52	2	30	15	1	33	406	128		
✓	0.00308	92	2	57	32	1	56	444	191		
	0.00533	127	3	104	47	1	95	461	248		
	0.00876	168	6	183	77	2	158	475	319		
	0.01372	180	19	✓	324	100	3	261	484	379	
198	0.6	5	1	563	129	3	438	491	424		
	2	8	1	967	160	3	752	495	446		
	4	11	1	1625	196	12					
	8	18	1	2621	236	51					

clusters at $q = 6.2$, $C = 0.014$ ($p = 0.5$), with 3 misclassifications. The number of support vectors was 40. This result (see Table 3) compares favorably with [4] and the q value is generated by our algorithm.

Table 3. The iris data is clustered with the parameters of $q = 6.2$ and $C = 0.014$. The left: shows data points with the contours as the mapping of the minimal hypersphere (shaded) and lines (dark area) representing the connectivity among data points in the contours. The right: 3 clusters with 3 misclassified.



For all of our datasets the final q value, for which all data points are

support vectors, is $\leq q_{max}$. Furthermore, our estimate of the q sequence length given by Eq. 8 has 83% accuracy with respect to the actual q sequence length for all datasets in Table 1. Monotonicity enforcement was not needed in our experiments.

Although choosing an appropriate C value is future work, we observe that the β values contain useful information to distinguish outliers from non-outliers. For example, if we compute the average β value of all data points across our q list and count the number of data points having larger β value than the average β value, then we can use the number as an approximate number of outliers in the data set. This experiment yields 107 outliers in the data set with $N = 500$ with $\frac{1}{107} = 0.009$ as a C value. This compares favorably with the 0.007 value suggested in [4].

4. Conclusions

We presented an algorithm that explores the Gaussian kernel width for support vector clustering. No algorithm had previously been proposed in the literature. The algorithm uses a secant-like method that generates an increasing sequence of kernel width values for constant outlier parameter value. Our estimate of the number of kernel width values depends on spatial characteristics of the data but not on the number of data points or the dimensionality of the dataset.

Future work includes testing additional high-dimensional datasets, such as ones from the UCI repository. We will also seek datasets for which the minimal sphere radius does not behave monotonically so we can evaluate the impact of our monotonicity enforcement techniques on SVC clustering accuracy. We plan to develop an algorithm for deciding which additional kernel width and outlier parameter values to investigate. We hope to develop a cluster labeling approach that works well in high dimensions and is efficient while providing good clusterings. Finally, it would be useful to embed our clustering procedures into visualization tools.

References

1. Sei-Hyung Lee and Karen Daniels, *Gaussian Kernel Width Exploration in Support Vector Clustering* **2004-009**, Technical Report, University of Massachusetts Lowell, (2004).
2. V. Estivill-Castro and I. Lee, *Automatic Clustering via Boundary Extraction for Mining Massive Point-data sets* In Proceedings of the 5th International Conference on Geocomputation, (2000).

3. Welzl Emo, *Smallest Enclosing Disks (Balls and Ellipsoids)* **LNCS 555** New Results and New Trends in Computer Science, (1991).
4. A. Ben-Hur, D. Horn, H. T. Siegelmann and V. Vapnik, *Support vector clustering* **2**, Journal of Machine Learning Research (2001).
5. Sammon JW Jr, *A nonlinear mapping for data structure analysis* **18** IEEE Transactions on Computers, (1969).
6. E. Hartuv and R. Shamir, *A clustering algorithm based on graph connectivity* (**200**), Information Processing Letters 76, (2000).
7. Anil K. Jain, N. M. Murty and P. J. Flynn, *Data Clustering: A Review* **Vol 31 (3)** ACMCS, (1999).
8. J. Yang, V. Estivill-Castro and S. K. Chalup, *Support Vector Clustering Through Proximity Graph Modeling* Proceedings, 9th International Conference on Neural Information Processing (ICONIP'02), (2002).
9. B. S. Everitt, S. Landau and M. Leese", *CLUSTER ANALYSIS* (2001).
10. V. N. Vapnik, *The nature of statistical learning theory*, (1995).
11. R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification*, (2001).
12. T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning*, (2001).
13. N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*, (2002).
14. D. Fasulo, *An Analysis of Recent Work on Clustering Algorithms* **01-03-02** Technical Report, University of Washington, (1999).
15. J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, (2001).
16. Erez Hartuv and Ron Shamir, *A clustering algorithm based on graph connectivity* **vol 76 4-6**, Information Processing Letters, (2000).
17. M. Ester, H. P. Kriegel, J. Sander and X. Xu, *A density-based algorithm for discovering clusters in large spatial databases with noise* Proc. 2nd Int. Conf. Knowledge Discovery and Data Mining(KD-96), (1996).
18. Istvan Jonyer, Lawrence B. Holder and Diane J. Cook, *Graph-Based Hierarchical Conceptual Clustering* **vol 10 1-2** International Journal on Artificial Intelligence Tools, (2001).
19. David Harel and Yehuda Koren, *Clustering spatial data using random walks* Knowledge Discovery and Data Mining (KDD'01), (2001).
20. V. Estivill-Castro and I. Lee, *Why so many clustering algorithms* **vol 4 (1)** SIGKDD Explorations, (2002).
21. V. Estivill-Castro and I. Lee, *Hierarchical clustering based on spatial proximity using delaunay diagram* **7a** In Proceedings of the 9th International Symposium on Spatial Data Handling, (2000).
22. J. Platt, B. Schölkopf, C. J. C. Burges and A. J. Smola, *Fast training of support vector machines using sequential minimal optimization* Advances in Kernel Methods - Support Vector Learning, (1999).
23. G. E. Forsythe, M. A. Malcolm and C. B. Moler, *Computer methods for mathematical computations* (1977).