The Support Vector Clustering Neural Network Used For Pattern Classification

Chin-Shyurng Fahn and Ming-Jui Kuo

Department of Computer Science and Information Engineering
National Taiwan University of Science and Technology
Taipei 106, Taiwan, Republic of China
csfahn@csie.ntust.edu.tw

ABSTRACT

In this paper, we present a novel neural network called Support Vector Clustering Neural Network (SVCNN). The theoretical foundation of the network is based on Support Vector Clustering (SVC). The training method of SVC is in a batch learning mode which has a drawback that the solution of the Lagrange multipliers is difficult to find when the training data become large. To overcome this drawback, we propose a mechanism, namely Support Vector Identifying Support Vector (SVISV), and develop its associated algorithm whose training method adopts an incremental learning mode. In each step, only one or few new data attend the SVC calculation. Then the data that just act as support vectors will remain to continuously attend the calculation in the next step. Until all the training data have been clustered, the learning process is terminated. The clustering result is further to construct a SVCNN. Using the mechanism SVISV, the SVCNN can determine whether an unknown data belongs to one cluster that has been already built in the network. The simulation outcomes reveal that our SVISV algorithm is slightly faster than the traditional SVC while support vectors are the minorities in a data set.

Keywords: support vector, support vector clustering, support vector identifying support vector, support vector clustering neural network.

1. Introduction

Support Vector Clustering (SVC) is a non-parametric clustering algorithm that has been proposed in 2001 [1], [2]. Unlike parametric [3] and hierarchical [4] clustering algorithms, SVC uses the Lagrange multipliers to obtain the support vectors (SVs) that can be employed to describe the contour of each cluster. Unfortunately, when the data set becomes large, solving the Lagrange formulation is more difficult. To surmount this, Ban and Abe presented a "Spatially Chunking Support Vector Clustering Algorithm" to speed up the learning process [5]. They segmented the original data set into several sub ones. Each of the sub data sets will be trained by SVC and then combined with the other individual results to acquire the global clustering one. In this paper, our goal is also to solve the large data problem arisen from using SVC. Based on the inherent property of SVC, we propose a mechanism called Support Vector Identifying Support Vector (SVISV), and then develop its associated algorithm such that we can alter the training method used in SVC from the batch learning mode into an incremental one. Furthermore, we extend this algorithm to create a novel neural network which is called Support Vector Clustering Neural Network (SVCNN). Such a network can determine whether an unknown data belongs to one of the clusters which have been built in the network. From the simulation outcomes, we can see that both SVC and the SVISV algorithm can obtain the same clustering result but only SVs remain in the latter. For a large data set, the SVISV algorithm can save much training time except that most of the training data are SVs. In addition to this, the SVCNN can successfully classify an unknown data.

2. Support Vector Clustering

The SVC scheme maps the training data from a data space into a high dimensional feature space by a nonlinear transform Φ. In the feature space, such transform seeks the smallest sphere of radius R which encloses all of the mapping data. Of the sphere, some mapping data lie on the surface, which play the role of SVs. The goal of SVC is to find the SVs that constitute the contour of a cluster. The mathematical formulation of SVC is briefly described as follows.

Let $\mathbf{X} = \{\mathbf{x}_i \mid i = 1,2,..., N\}$ be a data set in the data space $\Re^d$. The smallest sphere that we want to look for can be written as the following constraints:

$$\|\Phi(\mathbf{x}_i) - \mathbf{a}\|^2 \le R^2 \qquad i = 1,2,..., N \tag{1}$$

where $\|\bullet\|$ represents the Euclidian distance and $\mathbf{a}$ is the center of the sphere.

To solve Equation (1), the Lagrange formulation is included below,

$$L = R^2 - \sum_{i=1}^{N} (R^2 - \|\Phi(\mathbf{x}_i) - \mathbf{a}\|^2)\beta_i \quad (2)$$

where β_i is a Lagrange multiplier.

Following the SV approaches [6], [7], the SVC scheme adopts a kernel function to achieve a nonlinear transformation. In order to attain a tight contour of a cluster, the Gaussian kernel function is verified to use [8], which is expressed as

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-q\|\mathbf{x}_i - \mathbf{x}_j\|^2} \quad (3)$$

where q is the Gaussian window.

Figure 1 illustrates a data set in a data space mapped into a feature space via a nonlinear transform using the Gaussian kernel function. In the feature space, we seek the smallest sphere of radius R enclosing all of the mapping data. Some of the data lying on the surface of the sphere are SVs and they have the same distance from their locations to the center of the sphere. These SVs inversely mapped into the data space will constitute the contour of a cluster.

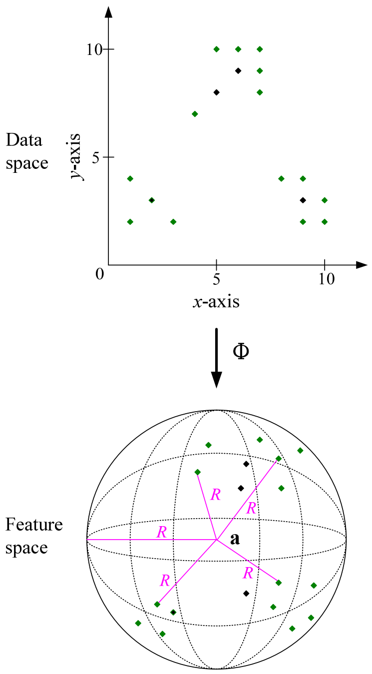

Figure 1 Illustration of a data set in a data space nonlinearly mapped into a feature space.

3. The Support Vector Clustering Neural Network

In this section, we will first introduce the mechanism SVISV. According to this, the SVISV algorithm is then developed, which can change the training method of the SVC from the batch learning mode into an incremental one. Moreover, this algorithm is extended to set up a novel network, SVCNN, which can determine whether an unknown data belongs to one cluster that has been built in the network.

3.1 The support vector identifying support vector

The training method of SVC is typically in a batch learning mode; that is, all of the training data are put together to solve their corresponding Lagrange multipliers simultaneously. However, when the training data become large, finding the solution of the Lagrange multipliers will be more and more difficult. In this paper, we propose an incremental learning mode algorithm to improve the above drawback. This algorithm is based on the mechanism SVISV which inspires us to further develop the SVCNN that can determine whether an unknown data belongs to one of the clusters which have been already built in the SVCNN. The following states the mechanism SVISV.

Assume a cluster C that has been established by SVC and its mapping data in form of a sphere of radius R in the feature space. To classify a new data, it is together with all the current data of C to perform the clustering by SVC and obtains a new radius R' in the feature space. According to the absolute difference value $|R\text{-}R'|$, we can judge whether the new data belongs to C. The principle of the decision rule is described as follows. The R' will be very close or equal to R if the new data locates inside C or exactly on its contour; that is, the new data is within C. In this case, the new data is mapped from a data space to a feature space, its mapping data then lies inside the sphere or on its surface automatically. In theory, the sphere can enclose this new data without changing its size. On the contrary, the R' will be larger than R when the new data lies outside C. In such a case, the new data does not belong to C. Therefore, the sphere must extend its size to enclose the mapping data. This mechanism is called SVISV that we can compare the absolute difference value between R and R' to determine whether the new data belongs to a given cluster. What follows demonstrates the mechanism SVISV.

A given cluster C is shown in Figure 2(a), which has been grouped by SVC. The rectangle and triangle represent two distinct existing data and the dashed line represents the contour, respectively. We use the cross to stand for the new data, which will be categorized into C by the SVC calculation in the following four cases. Figure 2(b) shows the new data lies inside C and Figure 2(c) shows the new data exactly lies on its contour. Each of these situations means that the new data belongs to C. It can be obviously seen that the new obtained radius R' is close to the radius R of C. Figure 2(d) shows the new data lies outside the original C and this data becomes a SV. The corresponding radius R' will be larger than R since the mapping data is outside the sphere. Figure 2(e) also shows the new data does not belong to C, but the rectangle is not still a SV since it lies inside the new extended cluster.

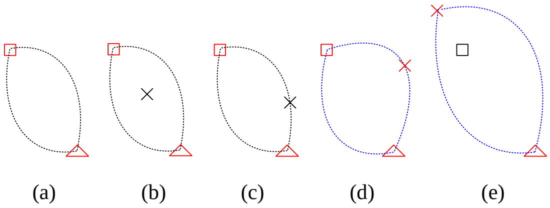

Figure 2 (a) A given cluster C resulting from SVC; (b)~(e) four different clustering results after a new data attending the SVC calculation.

3.2 The SVISV algorithm

We will decompose a data set $\mathbf{X}$ into some sub sets to sequentially perform clustering by SVC. Initially, only few data are selected to form a cluster C. Subsequently, one new data is retrieved from the remainder data in $\mathbf{X}$ and clustered by the SVC calculation in each step. Then only the data that are SVs will be kept to attend the clustering in the next step. Otherwise, the cross and rectangle that are not SVs will be deleted from the cluster C as Figures 2(b) and 2(e) respectively show. Until all the data in $\mathbf{X}$ have been selected to perform clustering by SVC, the algorithm will be terminated. Figure 3 depicts the SVISV algorithm in pseudo codes.

A critical problem in this SVISV algorithm is how to select one data to attend the clustering in each step. Actually, the data can be randomly selected without affecting the clustering result. But, in our simulation, we adopt a criterion that is the distance from a data point to the origin. For instance, the data can be selected from the farthest location down to the nearest one about the origin, and vice versa. Both of these two data input sequences will yield the same clustering result.

/* Given a training data set $\mathbf{X}$ consisting of N data points and an empty support vector set $\mathbf{S}$, if a data point $\mathbf{x}_k$ becomes a support vector, it will be denoted as $\mathbf{s}_k$ to add into $\mathbf{S}$. */

Input: $\mathbf{X} = \{\mathbf{x}_k \mid k = 1,2,\ldots,N\}$

Output: $\mathbf{S} \subseteq \mathbf{X}$

Initialize $\mathbf{S} = \{\mathbf{x}_1, \mathbf{x}_2\} = \{\mathbf{s}_1, \mathbf{s}_2\}$, $\mathbf{X} = \{\mathbf{x}_3, \mathbf{x}_4, \ldots, \mathbf{x}_N\}$

For i=3 to N

  Assign $\mathbf{x}_k$ to $\mathbf{s}_i$

  Put $\mathbf{s}_i$ into $\mathbf{S}$

  Cluster the $\mathbf{S}$ by the SVC calculation

  For j=1 to the cardinality of $\mathbf{S}$

    If ($\mathbf{s}_j$ is a support vector)

      Reserve $\mathbf{s}_j$ in the $\mathbf{S}$

    Else

      Remove $\mathbf{s}_j$ from the $\mathbf{S}$

    End if

  End for

End for

Figure 3 The SVISV algorithm.

3.3 The architecture of the support vector clustering neural network

Figure 4 shows the architecture of the SVCNN. This network is made up of four layers: the input layer, SVC layer, calculation layer, and output layer. The numbers of the input weightings of the output layer depends on the number of clusters after the training data have been conducted by the SVISV algorithm.

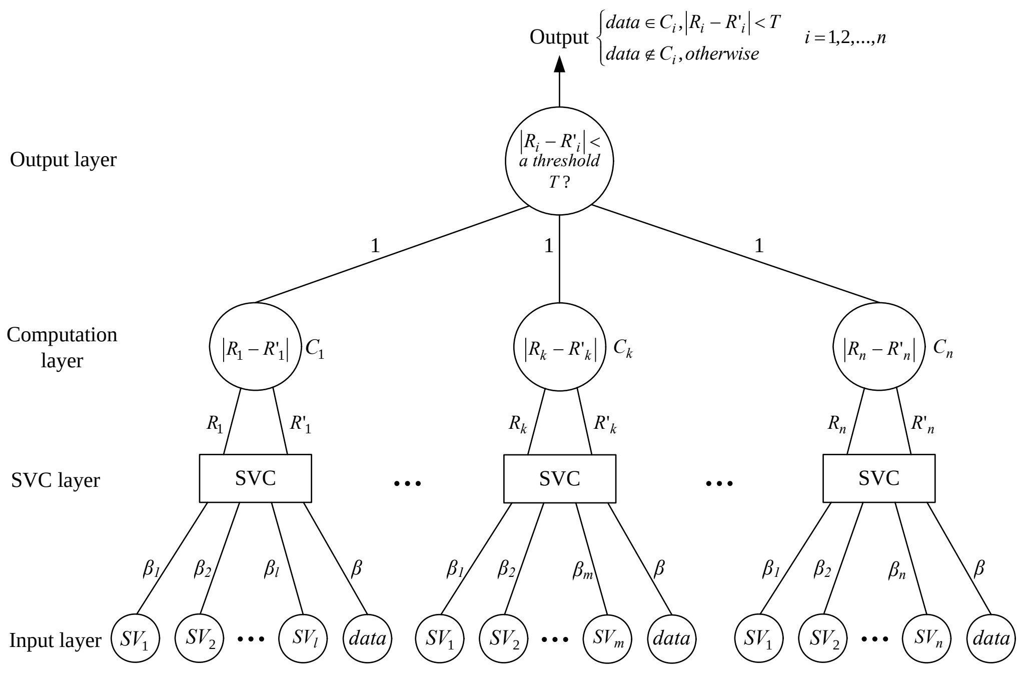

Figure 4 The architecture of the SVCNN.

In the input layer, the neurons labeled with "SV_j" mean the support vectors of each cluster through the execution of the SVISV algorithm. The corresponding weighting labeled with "β_j" obtained from training is a Lagrange multiplier for each SV. And the neurons labeled with "data" mean the same new data attending the SVC calculation for every cluster in the network. The second layer, namely the SVC layer, performs the SVC calculation with regard to the new data and the original members of a cluster. Such clustering will result in a new radius of the sphere, which is recorded in the output weighting labeled with "R'_i." Another output weighting of this layer is labeled with "R_i" representing the radius derived from the original members of each cluster after training. The third layer meaning the computation layer calculates the absolute difference value between R_i and R'_i which come from the SVC layer. The fourth layer is the output layer which sends out a message that whether the new data belongs to one of the clusters built in the network. The decision criterion is the value of $|R_i\text{-}R'_i|$ compared to a given threshold. If this value is greater than or equal to the threshold, it indicates that the sphere in the feature space has been extended. According to the mechanism SVISV, it implies the new data does not belong to the cluster. On the other hand, we can conclude that the new data belongs to one of existing clusters if the value of $|R_i\text{-}R'_i|$ is smaller than the threshold T.

The following is the training methodology of the SVCNN. The SVISV algorithm has grouped the original data set into n clusters, each of which only consists of SVs. We can regard each cluster as a new independent data set, and then adopt the SVC to re-cluster it. After clustering, every cluster will acquire its own parameters, including the Lagrange multiplier β_j corresponding to each SV and the radius R_i.

4. Simulation Results

In this section, we give two examples of simulations to demonstrate the effectiveness of the SVISV algorithm and its associated SVCNN. The data set shown in Figure 5(a) is the input data used in these simulations. Figure 5(b) is the clustering result obtained from SVC using the Gaussian kernel function with window q = 0.1; the dashed line stands for the contour of a cluster where the data point enclosed with a diamond represents a SV. The same clustering result as the above but only composed of SVs shown in Figure 5(c) can be received from the SVISV algorithm. The variations of radius R are shown in Figures 5(d) and 5(e), which are acquired by different input data sequences. In Figure 5(d), the new data is orderly pick up to attend the SVC calculation from the nearest location to the farthest one about the origin in each step. Figure 5(e) shows the result from the reverse order of picking up the new data depicted above. As we can see, no matter what the sequence is chosen, the final values of radius R are the same. In this simulation, both of them are 0.7612.

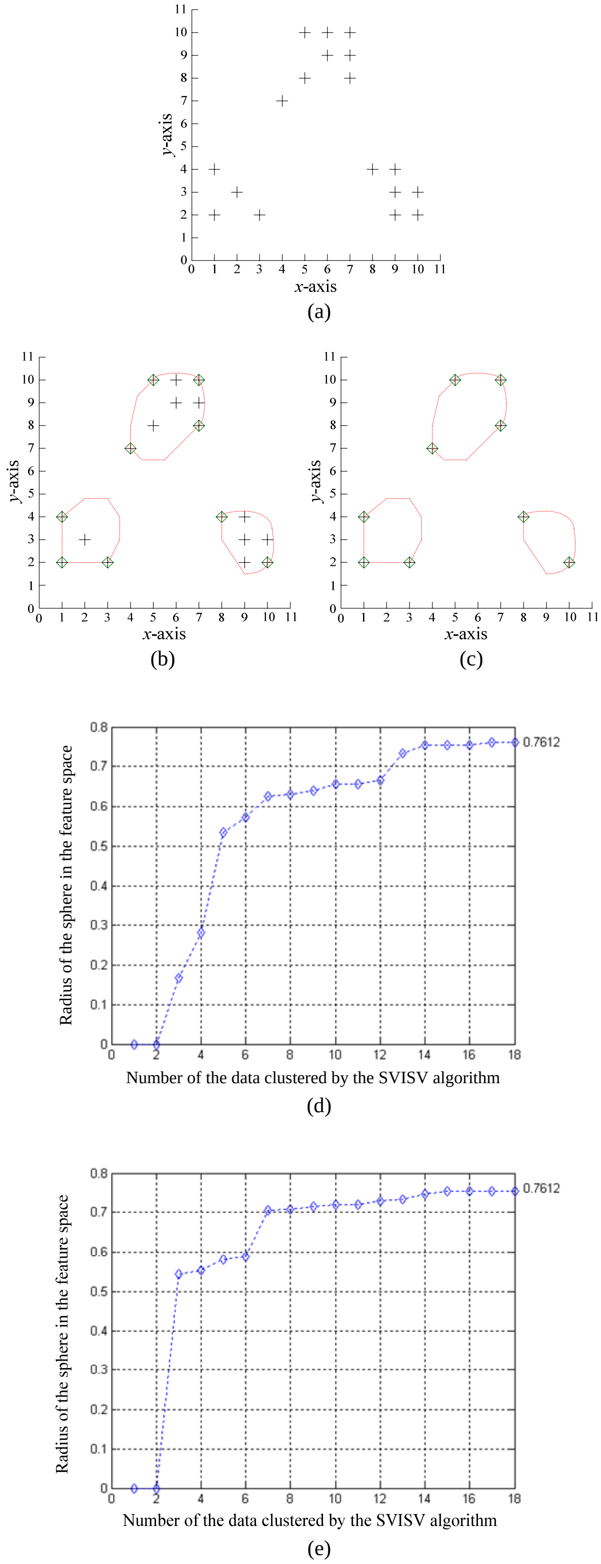

Figure 5 (a) A given data set; (b) the clustering result obtained from SVC; (c) the clustering result obtained from the SVISV algorithm; (d) the variations of R during the new data pick up in an increasing order according to the distance of its location from the origin; (e) the variations of R during the new data pick up in a decreasing order.

In the second simulation, we apply the aforementioned trained result to set up a SVCNN. Three clusters are built in the network and each of SVs is represented by one neuron, respectively. Table 1 lists the initial values of radius R of the three clusters in the SVCNN. After that, we use four test data to verify the classification ability of the SVCNN. Table 2 shows the classification results, where each row records the simulation outcome for one test data. We can observe that the test data belongs to which cluster by comparing the value of R'_i in Table 2 to that of R_i in Table 1. For example, say the second test data, (2,2), its absolute difference values associated with each cluster are |0.2830-0.2830|, |0.4362-0.6294|, and |0.2753-0.5750|, respectively. We can see that the value in the first term is zero, but the other two exceed zero. The SVCNN can classify this test data as Cluster 1 that has been built in the network.

Table 1
The Initial Value of Radius R_i for Each Cluster in the SVCNN

Radius \ Cluster	#1	#2	#3
R_i	0.2830	0.4362	0.2753

Table 2
The Classification Results of the Four Test Data Using the SVCNN

Test data \ R'_i \ Cluster	#1	#2	#3	Classification result
(6,8)	0.5766	0.4362	0.5482	Cluster 2
(2,2)	0.2830	0.6294	0.5750	Cluster 1
(6,5)	0.5348	0.5368	0.4644	New Cluster
(8,3)	0.5663	0.6266	0.2753	Cluster 3

All the above simulations are executed on an IBM T43 2668 OAV notebook equipped with an Intel Pentium IV 1.73GHz processor and 1.00 GB DDR DRAM. The development software is MATLAB 7.0, which is run in Microsoft Windows XP 2002 with Service Pack 2. The execution time resulting in Figure 5(b) by SVC is 0.112 second and that in Figure 5(c) by the SVISV algorithm is 0.087 second. From these outcomes, we can see that the SVISV algorithm is slightly faster than SVC while SVs are the minorities in a data set.

5. Conclusions

In this paper, we have presented a SVISV algorithm to improve the limitation in SVC; that is, when the training data become large, the solution of the Lagrange multipliers is difficult to find. Our proposed mechanism SVISV changes the training method in SVC from the batch learning mode to an incremental one. However, one problem is still not overcome that most of the data within a cluster are SVs. In this situation, when the last data was added to the cluster, the resulting set of the data will be almost the original one. It is caused by too many data that are just the SVs being reserved during each iterative step of the SVISV algorithm. Moreover, we extend the mechanism SVISV to develop a novel network called SVCNN. It can determine whether an unknown data belongs to one of the clusters that have been already built in the network. In the future, our goal is to make the SVCNN can recognize an object detected in real scene images.

REFERENCES

[1] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik, “A support vector clustering method,” in *Proceedings of the 15th International Conference on Pattern Recognition*, vol. 2, pp. 724-727, 2000.

[2] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik, “Support vector clustering,” *Journal of Machine Learning Research*, vol. 2, pp. 125-137, 2001.

[3] E. Gose, R. Johnsonnaugh, and S. Jost, *Pattern Recognition and Image Analysis*, Prentice Hall, Upper Saddle River, NJ, 1996.

[4] A. B. Geva, “Hierarchical unsupervised fuzzy clustering,” *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 6, pp. 723-733, 1999.

[5] T. Ban and S. Abe, “Spatially chunking support vector clustering algorithm,” in *Proceedings of the IEEE International Joint Conference on Neural Networks*, vol. 1, pp. 25-29, 2004.

[6] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, NY, 1995.

[7] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, NY, 1998.

[8] D. Tax and R. Duin, “Support vector domain description,” *Pattern Recognition Letters*, vol. 20, pp. 1191-1199, 1999.