ACSIJ Advances in Computer Science: an International Journal, Vol. 4, Issue 6, No.18 , November 2015
ISSN : 2322-5157
www.ACSIJ.org

ACSIJ
WWW.ACSIJ.ORG

Density Weighted Core Support Vector Machine

Shuxia Lu[1,*], Chenxu Zhu[2] and Caihong Jiao[1]

[1] Key Lab. of Machine Learning and Computational Intelligence,
College of Mathematics and Information Science, Hebei University
Baoding, Hebei 071002,China
cmclusx@126.com

[2] College of Science, Northwest Agriculture & Forestry University,
Yangling, Shanxi 712100, China
1441571065@qq.com

[1] Key Lab. of Machine Learning and Computational Intelligence,
College of Mathematics and Information Science, Hebei University
Baoding, Hebei 071002,China
1039877570@ qq.com

Abstract

Core Vector Machine (CVM) can be used to deal with large data sets classification problem, but CVM do not consider the density distribution of the data. In order to obtain the optimal description of the data, we propose a density weighted core support vector machine (DWCVM). In the proposed DWCVM, the relative density of each data point is based on the density distribution of the target data using the *k*-nearest neighbor (*k*-NN) approach. Experimental results on several benchmark data sets show that the performance of DWCVM is much better than CVM.

Keywords: *minimum enclosing ball, core set, support vector domain description, density, core vector machine.*

1. Introduction

Classification is a fundamental task in machine learning, data mining and pattern recognition. Prominent methods include support vector machine (SVM) [1], Kernel Density Estimator (KDE) [2], Support Vector Data Description (SVDD) [3], Small Sphere and Large Margin approach [4] for one-class classification and novelty detection, and so on. These methods involve solving the corresponding quadratic programming (QP) problems [5], which heavily limits the applicability of these methods for a large dataset.

In order to circumvent this drawback, many endeavors have been made to develop various techniques for scaling up these QP solvers. Typical techniques include chunking or some complicated decomposition methods such as the SMO algorithm [6]. Core Vector Machine (CVM) [7, 8] was proposed by Tsang et al. (2005), Tsang et al. proposed the core vector machine (CVM) by utilizing an approximation algorithm for the minimum enclosing ball (MEB) problem in computational geometry, the CVM algorithm achieves an asymptotic time complexity that is linear in N and a space complexity that is independent of N, where N is the size of the training patterns.

Inspired by [9] we propose a density weighted core support vector machine (DWCVM). In the proposed DWCVM, the relative density of each data point is based on the density distribution of the target data using the *k*-nearest neighbor (*k*-NN) approach. An optimal description of the data can be obtained by incorporating the weight into the search for using SVDD. Experimental results on several data sets demonstrate the effectiveness of DWCVM.

The rest of the paper is organized as follows. Section 2 reviews MEB, SVDD and GCVM. Section 3 describes the proposed DWCVM in detail. Experimental results are reported in Section 4. Concluding remarks are presented in Section 5.

2. Background

2.1 Standard MEB

The MEB problem aims to finding a smallest ball to enclose all training data defined by the sample set $S=\{x_i \mid x_i \in \mathbb{R}^n,\ i=1,\cdots,N\}$. The smallest ball denoted as $B(c,R)$ with center c and radius R . It is determined by solving

$$\min_{R,c} R^2 \quad s.t. \|\phi(x_i) - c\| \le R, \quad 1 \le i \le N. \tag{1}$$

which is similar to a one-class SVDD with hard margin and is called the standard MEB here. The corresponding dual of (1) is the following QP problem

$$\max_{\alpha} \boldsymbol{\alpha}^T diag(\mathbf{K}) - \boldsymbol{\alpha}^{\mathbf{T}}\mathbf{K}\boldsymbol{\alpha} \quad s.t. \boldsymbol{\alpha}^T \mathbf{1} = 1, \quad 1 \le i \le N. \tag{2}$$

Where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2 \ldots, \alpha_N]^T \ge \mathbf{0}$ is the Lagrangian multipliers, $\mathbf{1} = [1,1,\ldots,1]^T$ is an *N*-dimensional vector, and $\mathbf{K} = [\phi(x_i)^T \phi(x_j)]_{N\times N} = [k(x_i, x_j)]_{N\times N}$ is the corresponding $N \times N$ kernel matrix with the term $k(x_i, x_j)$ denoting a kernel function.

2.2 SVDD

Tax and Duin (2004) presented the Support Vector Data Description (SVDD) which can obtain a spherically shaped boundary and the boundary that can be used to enclose normal data (similar to an enclosing sphere) and detect novel data or outliers (i.e. outside the enclosing sphere). The primal problem of SVDD is:

$$\min_{R,c,\xi_i} R^2 + C\sum_{i=1}^{N} \xi_i \quad s.t. \|\phi(x_i) - c\|^2 \le R^2 + \xi_i, \quad \forall i = 1,\ldots,N. \tag{3}$$

where *C* is regularized parameters which control the volume of boundary and the errors, and *c* and *R* respectively the center and radius of the sphere, denoted as $B(\mathrm{c}, \mathrm{R})$. The corresponding dual of (3) is

$$\max_{\alpha} \boldsymbol{\alpha}^T diag(\mathbf{K}) - \boldsymbol{\alpha}^{\mathbf{T}}\mathbf{K}\boldsymbol{\alpha} \quad s.t. \boldsymbol{\alpha}^T \mathbf{1} = 1, \quad \mathbf{0} \le \boldsymbol{\alpha} \le \mathbf{C}. \tag{4}$$

where $\boldsymbol{\alpha} = \left[\alpha_1, \alpha_2, \ldots, \alpha_N\right]^T$ are the Lagrange multipliers.

2.3 The generalized core vector machine (GCVM）

The generalized core vector machine (The generalized CVM, GCVM) algorithm is proposed in [8]. The GCVM algorithm is much faster and can handle much larger datasets than existing SVM implementations. The generalized CVM algorithm can be used with any linear/nonlinear kernel and can also be applied to kernel methods such as SVR and the ranking SVM.

The GCVM utilizes an approximation algorithm for the center constrain minimum enclosing ball (CC-MEB) problem, which will be briefly introduced as follows:

The center and radius of a ball $B(\mathbf{c}, R)$ are denoted by $\mathbf{c}_B$ and r_B, respectively. Given an $\varepsilon > 0$, a ball $B(\mathbf{c}, (1+\varepsilon)R)$ is an $(1+\varepsilon)$ -approximation of $MEB(S)$ if $R \le r_{MEB(S)}$ and $S \subset B(\mathbf{c}, (1+\varepsilon)R)$. $\varphi : x_i \to \varphi(x_i)$ denotes the feature map associated with a given kernel *k*, and $B(\mathbf{c}, R)$ is the desired MEB in the kernel-induced feature space Γ.

The MEB problem finds the smallest ball containing all $\varphi(x_i) \in S$ in the feature space. In this section, we first augment an extra $\delta_i \in R$ to each $\varphi(x_i)$, forming $\begin{bmatrix} \varphi(x_i) \\ \delta_i \end{bmatrix}$. Then, we find the MEB for these augmented points, while at the same time constraining the last coordinate of the ball's center to be zero (i.e., of the form $\begin{bmatrix} \mathbf{c} \\ 0 \end{bmatrix}$). The primal form of the center constrain minimum enclosing ball (CC-MEB) problem can be formulated as

$$\min \quad R^2 \quad s.t. \quad \|\varphi(x_i) - \mathbf{c}\|^2 + \delta_i^2 \le R^2, \quad i = 1,\cdots,N. \tag{5}$$

The corresponding dual of (5) is the following QP problem

$$\max \quad \boldsymbol{\alpha}^T (diag(\mathbf{K}) + \boldsymbol{\Delta}) - \boldsymbol{\alpha}^T \mathbf{K}\boldsymbol{\alpha} \quad s.t. \quad \boldsymbol{\alpha}^T \mathbf{1} = 1, \quad \boldsymbol{\alpha} \ge \mathbf{0}. \tag{6}$$

where $K = [k(x_i, x_j)] = [\varphi(x_i)^T \varphi(x_j)]$ is the corresponding kernel matrix, and

$$\boldsymbol{\Delta} = [\delta_1^2, \cdots, \delta_N^2]^T \ge \mathbf{0}. \tag{7}$$

From the optimal $\boldsymbol{\alpha}$ solution of (6), we can recover *R* and $\mathbf{c}$ as

$$R = \sqrt{\boldsymbol{\alpha}^T (diag(\mathbf{K}) + \boldsymbol{\Delta}) - \boldsymbol{\alpha}^T \mathbf{K}\boldsymbol{\alpha}} \tag{8}$$

$$\mathbf{c} = \sum_{i=1}^{N} \alpha_i \varphi(x_i). \tag{9}$$

The squared distance between the center $\begin{bmatrix}\mathbf{c}\\0\end{bmatrix}$ and any point $\begin{bmatrix}\varphi(x_l)\\ \delta_l\end{bmatrix}$

$$\|\varphi(x_l)-\mathbf{c}\|^2+\delta_l^2=\|\mathbf{c}\|^2-2(\mathbf{K}\boldsymbol{\alpha})_l+k_{ll}+\delta_l^2. \quad (10)$$

which does not depend explicitly on the feature map φ.

Because of the constraint $\boldsymbol{\alpha}^T\mathbf{1}=1$ in (6), an arbitrary multiple of $\boldsymbol{\alpha}^T\mathbf{1}$ can be added to the objective without affecting its solution. In other words, for an arbitrary $\eta\in\square$, (6) yields the same optimal as

$$\begin{aligned}&\max \quad \boldsymbol{\alpha}^T(diag(\mathbf{K})+\boldsymbol{\Delta}-\eta\mathbf{1})-\boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha}\\ &s.t. \quad \boldsymbol{\alpha}^T\mathbf{1}=1, \quad \boldsymbol{\alpha}\geq\mathbf{0}.\end{aligned} \quad (11)$$

Hence, any QP problem of the form (11), with the condition (7), can also be regarded as a special MEB problem, called center constrained MEB, i.e. CC-MEB. As pointed out by Tsang et al., CC-MEB can be approximately solved with the asymptotic linear time complexity $O(N)$ and its space complexity independent of N for large datasets by using the generalized core vector machine.

The GCVM algorithm is introduced as follows:

The GCVM algorithm is shown in Algorithm 1. Here, the core set, the ball's center, and radius at the tth iteration are denoted by S_t, $\mathbf{c}_t$, and R_t respectively. The GCVM algorithm requires the input of a termination parameter ε.

Algorithm 1. GCVM

1) Initialize ε, $t=0, S_t, \mathbf{c}_t, R_t$.
2) Update the core set: if there is no training pattern that falls outside the ball $B(\mathbf{c}_t,(1+\varepsilon)R_t)$ in the corresponding feature space, $S=S_t$.
3) Find $\mathbf{z}$ such that it is the farthest away from $\mathbf{c}_t$ in the corresponding feature space and set $S_{t+1}=S_t\bigcup\{\mathbf{z}\}$.
4) Find the new MEB: $B(\mathbf{c}_{t+1},R_{t+1})$.
5) Set $t=t+1$, and go to step 2.

3. Density weighted core support vector machine

To accurately reflect the characteristics of the target data set, we propose a density weighted core support vector machine (DWCVM). In the proposed DWCVM, the relative density of each data point is based on the density distribution of the target data using the *k*-nearest neighbor (*k*-NN) approach. The distance between x_i and the *k*th nearest neighbor of x_i is denoted as $d(x_i,x_i^k)$; where x_i^k is the *k*th nearest neighbor of data point x_i. Using *k*-NN distance, the density weight of data point x_i is defined as:

$$w_i=1-\frac{d(x_i,x_i^k)}{\max_{j\in train\ set} d(x_j,x_j^k)} \quad (12)$$

Density weight measures the relative density based on the density distribution of the target data by comparing the *k*-NN distance of each data point with the maximum *k*-NN distance of the dataset. Density weight falls within the range $0\leq w_i\leq 1$.

To measure the density weight in feature space, can use the kernel function to map data into high dimensional space. The distribution of the data in feature space may be different from the original data distribution. In order to obtain a more appropriate description, we estimate the density weight in real space.

According to the density weight estimation method in (12), a data point located in a comparatively high-density area is close to its neighbors, so the distance between that data point and its *k*th nearest neighbor decreases, and eventually the density weight will become larger. In relatively low-density areas, data points are far from each other, so the density weight value will be low.

To apply the density weight, the objective function is defined as follows:

$$\begin{aligned}&\min_{R,c,\xi_i} R^2+C\sum_{i=1}^{N}w_i\xi_i\\ &s.t.\ \|\phi(x_i)-c\|^2\leq R^2+\xi_i, \quad \forall i=1,\ldots,N.\end{aligned} \quad (13)$$

We impose the weight w_i on each data point x_i. The data points in high-density regions receive a larger weight, so the effect of the slack variable is compounded. Therefore, to minimize the objective function, the spherical description will shift toward the high-density regions. On the other hand, with decreasing weight in relatively sparse areas, the influence of each data point will be reduced and

there is no pressure to keep data lying outside the spherical description.

By introducing the Lagrangian function for (13), and let partial differentiation of the Lagrangian function is equal to 0, we have the Wolf dual form

$$\begin{aligned} &\max_{\beta} \sum_{i=1}^{N} \beta_i \, \mathrm{k}(x_i, x_i) - \sum_{i,j=1}^{N} \beta_i \beta_j \, \mathrm{k}(x_i, x_j) \\ &s.t. \quad 0 \le \beta_i \le w_i C, \\ &\quad \sum_{i=1}^{N} \beta_i = 1, \quad \forall i = 1, 2, \ldots, N. \end{aligned} \tag{14}$$

Notice that, the upper bounds for Lagrange multipliers $\beta_i, i = 1, \ldots, N$ are no longer the same. Instead, each of them is respectively controlled by the corresponding weight. The primal variables can be recovered from the optimal β as

$$c = \sum_{i=1}^{N} \beta_i \phi(x_i), \quad R = \sqrt{\boldsymbol{\beta}^T diag(\mathbf{K}) - \boldsymbol{\beta}^{\mathbf{T}} \mathbf{K} \boldsymbol{\beta}}. \tag{15}$$

Therefore, by introducing density weight into the search for the optimal description of the dataset, can shift its description boundary to dense areas. In our proposed DWCVM, update core set by using density weight MEB method, and then train the core set by using SVM algorithm.

The DWCVM algorithm is introduced as follows:

Algorithm2. DWCVM

Step 1): Initialize $\varepsilon, t = 0, S_t, \mathbf{c}_t, R_t$.

Step 2): Update the core set: Terminate if there is no training point z such that $\phi(\mathrm{z})$ falls outside the $(1+\varepsilon)$ -ball $B(c_t, (1+\varepsilon)R_t)$ in the corresponding feature space, $S = S_t$.

Step 3): Find z such that $\phi(\mathrm{z})$ is furthest away from c_t in the corresponding feature space and set $S_{t+1} = S_t \bigcup \{z\}$.

Step 4): Find the new MEB: The c_{t+1} and R_{t+1} are computed by (15) .

Step 5): Set $t = t + 1$ and go back to step 2).

Step 6): Train the core set using SVM algorithm.

4. Experimental results

In this section, we compared the proposed algorithm DWCVM with CVM on several datasets for performance evaluation. In all experiments, the QP solver is adopted to solve the QP problem and the Gaussian function $k(x, y) = \exp(-\|x - y\|^2 / h)$ is taken as the kernel function, where h is the kernel parameter of the Gaussian kernel. In all experiments, the kernel parameter is $s^2/4$, s is the mean squared norm of the training data. All the experiments were carried out on a 3.1 GHz Pentium Core(TM) machine with 8GB RAM, running on the Matlab7.8 platform.

4.1 Data sets

The numbers of attributes, samples, positive samples and negative samples are shown in Table 1. The MiniBooNE dataset is used to distinguish electron neutrinos (signal) from muon neutrinos (background). The skin Segmentation dataset is constructed over B, G, R skin and Nonskin dataset is generated using skin textures from face images of diversity age, gender and race people.

We separately adopt the testing accuracy and geometric mean accuracy to evaluate the performance of algorithms. Considering the imbalanced nature of the training datasets, the geometric mean accuracy can be used. The geometric mean accuracy is defined as $g = \sqrt{a^+ \cdot a^-}$, where a^+ and a^- is computed by using Eq. (16). The measure takes into consideration the classification results on both positive and negative classes.

$$a^+ = \frac{\#\, positive\ sample\ correctly\ classified}{\#total\ \ positive\ sample\ classified} \times 100\%,$$

$$a^- = \frac{\#negative\ sample\ correctly\ classified}{\#total\ negative\ sample\ classified} \times 100\%. \tag{16}$$

Table 1: Summary of the data sets

Data sets	Attributes	Samples	Positive Samples	Negative Samples
MiniBooNE	51	130064	36499	93565
Spambase	58	4602	1813	2788
Skin	4	245057	50859	194198
Codrna	9	488565	162855	325710
Shuttle	10	58000	45586	12414
Sat	37	6435	3594	2841
Digit	65	5620	1697	3923

4.2 Performance evaluation

Experiment 1: In this experiment, we try to analyze the influence of the approximation parameter ε in the proposed DWCVM on the shuttle dataset. The percent 50

of samples are randomly selected as training data sets and the rest of the samples are used for testing data sets. The experimental results are listed in Table 2. From Table 2, we can see that as ε decreases, the geometric mean accuracy and the testing accuracy become higher, and the training time and the testing time become much more. Therefore, setting $\varepsilon = 1e-4$ is acceptable in the trade-off of the training speed and the classification accuracy for most cases.

Table 2: Influence of parameter ε on DWCVM

ε	g Accuracy	Testing Accuracy	Training Time (s)	Testing Time (s)
1e-2	82.54	85.21	0.11	1.24
1e-3	93.61	94.89	0.18	1.41
1e-4	94.25	97.12	0.25	1.74
1e-5	96.34	98.11	0.33	2.45
1e-6	97.21	98.46	0.71	3.21
1e-7	98.21	99.15	1.54	6.23

Table 3: Accuracy comparisons of DWCVM and CVM

Data sets	DWCVM		CVM	
	g Accuracy	Testing Accuracy	g Accuracy	Testing Accuracy
MiniBooNE	74.36	75.42	71.65	73.23
Spambase	75.64	76.24	75.25	74.10
Skin	98.57	98.87	95.45	92.62
Codrna	76.32	76.75	70.89	69.89
Shuttle	91.23	95.65	89.23	93.78
Sat	90.12	95.32	89.24	89.24
Digit	89.21	94.11	88.54	92.03

Table 4: Time comparisons of DWCVM and CVM

Data sets	DWCVM		CVM	
	Training Time (s)	Testing Time (s)	Training Time (s)	Testing Time (s)
MiniBooNE	28.03	20.11	13.10	16.56
Spambase	3.02	0.38	1.18	0.12
Skin	15.89	4.70	14.27	3.58
Codrna	27.32	4.65	37.84	9.25
Shuttle	1.56	0.20	1.71	0.21
Sat	8.33	0.80	4.31	1.23
Digit	31.10	2.01	42.54	2.22

Experiment 2: In this experiment, we compared the performance of DWCVM and CVM. For MiniBooNE and Skin Segmentation datasets, the percent 70 of samples are randomly selected as training data sets and the rest of samples are used for testing data sets. For the other data sets, the percent 50 of samples are randomly selected as training data sets and the rest of the samples are used for testing data sets. Table 3 and Table 4 illustrate the experimental results. The geometric accuracy and the testing accuracy comparisons of DWCVM and CVM are shown in Table 3. The training time and testing time comparisons of DWCVM and CVM are shown in Table 4. From Table 3, we can see that both the geometric accuracy and the testing accuracy of DWCVM are better than that of CVM. From Table 4, we can see that times of DWCVM and CVM are similar.

5. Conclusions

In order to consider the density distribution of the data, and deal with large data sets classification problem, we proposed the density weighted core support vector machine (DWCVM). In our proposed DWCVM, update core set by using density weight MEB method, and then train the core set by using SVM algorithm. The relative density of each data point is based on the density distribution of the target data using the *k*-nearest neighbor (*k*-NN) approach. Aims to accurately reflect the data density distribution of a target dataset with the weight of each data point based on relative density. This method prioritizes data points in high-density regions, and eventually the optimal description shifts to these regions. The application of a density measure for the dataset is beneficial for outlier detection, and generates a better performance. Experimental results on several data sets demonstrate the effectiveness of DWCVM.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (61170040), by the Natural Science Foundation of Hebei Province (F2015201185, F2013201220).

References

[1] C. C. Chang, and C. J. Lin, "Training v-support vector classifiers: theory and algorithms", Neural Computation, Vol.14, 2002, pp. 43-54.

[2] M. D. Marizio, and C. C. Taylor, "Kernel density classification and boosting: an L_2 analysis, Statistics and Computing", Vol. 15, No. 2, 2005, pp.13-123.

[3] D. M.J. Tax, and R. P. W. Duin, "Support vector data description", Machine Learning, Vol. 54, No. 1, 2004, pp: 45-66.

[4] M. R. Wu, and J. P. Ye, "A small sphere and large margin approach for novelty detection using training data with outlier", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 31, No. 11, 2009, pp. 2088-2092

[5] W. J. Hu, F. L. Chung, and S. T. Wang, "The maximum vector angular margin classifier and its fast training on large datasets using a core vector machine", Neural Networks, Vol. 27, 2012, pp. 60-73.

[6] N. Takahashi, and T. Nishi, "Rigorous proof of termination of SMO algorithm for support vector machines", IEEE Transaction on Neural Networks, Vol. 16, No. 3, 2005, pp. 774-776.
[7] I. W. Tsang, J. T. Kwok, and P. M. Cheung, "Core Vector Machine: Fast SVM training on very large data sets", Journal of Machine Learning Research, Vol. 6, 2005, pp. 363-392.
[8] I. W. Tsang, J. T. Kwok, and J. M. Zurada, "Generalized core vector machines", IEEE Transactions on Neural Networks, Vol. 17, No. 5, 2006, pp. 1126-1140.
[9] C. Myungraee, S. K. Jun, and B. Jun-Geol, "Density weighted support vector data description", Expert Systems with Applications, Vol. 41, 2014, pp. 3343–3350.

Shuxia Lu is a professor of the Faculty of Mathematics and Information Science, Hebei University. Received the B.Sc. and M.Sc. degrees in Mathematics from Hebei University, Baoding, China, in 1988 and 1991, respectively, and the Ph.D. degree from Hebei University, Baoding, China, in 2007. Her main research interests include machine learning and computational intelligence, SVMs.

Chenxu Zhu has been a B.Sc. degree candidate in College of Science from Northwest Agriculture & Forestry University, Yangling, China. Her research interests include computational intelligence.

Caihong Jiao received the M.Sc. degree in Applied Mathematics from Hebei University, Baoding, China, in 2015. Her research interests include Machine Learning.