Letters

Density-Induced Support Vector Data Description

KiYoung Lee, Dae-Won Kim, Kwang H. Lee, and Doheon Lee

***Abstract*—The purpose of data description is to give a compact description of the target data that represents most of its characteristics. In a support vector data description (SVDD), the compact description of target data is given in a hyperspherical model, which is determined by a small portion of data called support vectors. Despite the usefulness of the conventional SVDD, however, it may not identify the optimal solution of target description especially when the support vectors do not have the overall characteristics of the target data. To address the issue in SVDD methodology, we propose a new SVDD by introducing new distance measurements based on the notion of a relative density degree for each data point in order to reflect the distribution of a given data set. Moreover, for a real application, we extend the proposed method for the protein localization prediction problem which is a multiclass and multilabel problem. Experiments with various real data sets show promising results.**

***Index Terms*—Data domain description, density-induced support vector data description (D-SVDD), one-class classification, outlier detection, support vector data description (SVDD).**

I. Introduction

The purpose of data description (also called one-class classification) is to give a compact description of a set of data referred to as target data. It is usually used for outlier detection (the detection of uncharacterized objects in a target data set) or for conventional multiclass classification problems especially where some of the classes are undersampled [1]. Several approaches such as density estimation approach [2], [3], boundary prediction approach [4], [5], and reconstruction approach using clustering methods [1], [6] have been used to finding out the compact description. When negative data that should be rejected in a compact description are available, conventional multiclass classification methods were also used for finding the compact description of target data [7].

Recently, Tax and Duin [1], [5] have invented support vector data description (SVDD) which was inspired by support vector machines (SVMs) [8]–[10]. In an SVDD [1], [5], the compact description of target data is given as a hypersphere $(\mathbf{a}, R)$ with minimum volume containing most of the target data; the objective function O

$$O = R^2 + C^+ \sum_{i=1}^{n} \xi_i \tag{1}$$

subject to $(\mathbf{x}_i - \mathbf{a}) \cdots (\mathbf{x}_i - \mathbf{a}) \leq R^2 + \xi_i$ and $\xi_i \geq 0$, where n is the total number of target data and the parameter $C^+(> 0)$ gives the tradeoff between volume of a hypersphere and the number of errors [5]. Analogous to SVMs [9], [11], [12], the data description is described by a few training data called support vectors [1], [5], and the kernel trick [9] is utilized to find a more flexible data description in a high-dimensional feature space [5].

Despite the usefulness of an SVDD [1], [5], however, the conventional SVDD (C-SVDD) has limitations to reflect overall characteristics of a target data set with respect to its density distribution. In the C-SVDD, as mentioned before, the small portion of data called support vectors fully determine the solution of target data description, whereas all of the nonsupport vectors have no influence on the solution of target description in the C-SVDD, regardless of the density distribution. However, the region around a nonsupport vector with higher density degree should be included in a compact description rather than other regions in order to more correctly identify the data description of the given data set. Hence, the solution solely based on the support vectors, without considering the density distribution, can miss the optimal solution.

To address the previous problem in the C-SVDD, we propose a density-induced SVDD (D-SVDD) to reflect the density distribution of a target data set by introducing the notion of a relative density degree for each data point. By using density-induced distance measurements both for target data and for negative data based on the proposed relative degrees, the D-SVDD can shift the center of hypersphere to the denser region based on the assumption that there are more data points in a denser region. Moreover, for a real application, we extend the D-SVDD to the protein localization prediction problem which is a multiclass and multilabel challenge.

The structure of this letter is organized as follows. In Section II, we introduce two methods for extracting relative density degrees, and theoretically formulate the proposed D-SVDD with new distance measurements based on the degrees. Section III highlights the potentials of the proposed approach through various experimental examples. In Section IV, we apply the proposed method to a protein localization prediction problem. Concluding remarks are presented in Section V.

II. Density-Induced SVDD

A. Relative Density Degree and Density Induced Distance

To reflect the density distribution into the search of optimal solutions of SVDD, we first introduce the notion of relative density degrees. The relative density degree for a data point represents how dense the region of the corresponding data point is compared to other regions in a given data set. Even though several approaches can be applied to extract relative density degrees, in this letter, we propose two methods to extract the relative density degree for each data point from a data set using a nearest neighborhood approach (Method I) and a Parzen-window approach (Method II).

In Method I, by using $d(\mathbf{x}_i, \mathbf{x}_i^K)$, the distance between $\mathbf{x}_i$ and $\mathbf{x}_i^K$ (the Kth nearest neighborhood of $\mathbf{x}_i$), and the mean distance of Kth nearest neighborhoods of all target data, $\Im^K$, the relative density degree ρ_i for $\mathbf{x}_i$ is defined by

$$\rho_i = \exp\left\{\omega \times \frac{\Im^K}{d(\mathbf{x}_i, \mathbf{x}_i^K)}\right\}, \qquad i = 1, \cdots, n \tag{2}$$

where $\Im^K = (1/n)\sum_{i=1}^{n} d(\mathbf{x}_i, \mathbf{x}_i^K)$, n is the number of data in a target class, and $0 \leq \omega \leq 1$ is a weighting factor. Note that this method

Manuscript received February 20, 2006; revised June 27, 2006. This work was supported by the National Research Laboratory under Grant 2005-01450 and the Korean Systems Biology Research under Grant 2005-00343 from the Ministry of Science and Technology.

K. Lee and D. Lee are with the Department of BioSystems, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 305-701, Republic of Korea (e-mail: kylee@biosoft.kaist.ac.kr; dhlee@biosoft.kaist.ac.kr).

D.-W. Kim is with the School of Computer Science and Engineering, Chung-Ang University, Seoul 156-756, Republic of Korea (e-mail: dwkim@cau.ac.kr).

K. H. Lee is with the Department of Electrical Engineering and Computer Science and the Department of BioSystems, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 305-701, Republic of Korea (e-mail: khlee@biosoft.kaist.ac.kr).

Digital Object Identifier 10.1109/TNN.2006.884673

reports higher relative density degree ρ_i for the data point in a higher density region: The data point with lower distance from its Kth nearest neighborhood has a higher ρ_i value.

In Method II, the relative density degree ρ_i is defined by the exponentially weighted Parzen-window density that is normalized by $\Im$ (the mean of all the Parzen-window density degrees); that is, ρ_i is defined by

$$\rho_i = \exp\left\{\omega \times \frac{\mathrm{Par}(\mathbf{x}_i)}{\Im}\right\}, \qquad i = 1, \cdots, n \tag{3}$$

where $\mathrm{Par}(\mathbf{x}_i) = (1/n)\sum_{j=1}^{n}(1/\sqrt{(2\pi)^d s})\exp(-(1/2s)(\mathbf{x}_i - \mathbf{x}_j)^2)$, $\Im = (1/n)\sum_{i=1}^{n}\mathrm{Par}(\mathbf{x}_i)$, d is the feature dimension of input data, and s is the smoothing parameter of the Parzen-window density [13]. Note that this method also reports higher ρ_i for the data point with higher Parzen-window density regarding to the mean of Parzen-window densities of all data.

After calculating the relative density degrees, to incorporate the degrees into searching the optimal description in the SVDD methodology, we propose new geometric distance measures called density-induced distance measures. First, to incorporate the density degrees of target data into searching the optimal description in an SVDD, we propose a new geometric distance called a positive density-induced distance. Suppose that each target data point can be represented as $(\mathbf{x}_i, \rho_i)$, where ρ_i is the relative density degree of $\mathbf{x}_i$. We define a *positive density-induced distance* δ_i^+ between target data point $\mathbf{x}_i$ and the center of a hyperspherical model $(\mathbf{a}, R)$ of a target data set as

$$\delta_i^+ \equiv \{\rho_i(\mathbf{x}_i - \mathbf{a}) \cdot (\mathbf{x}_i - \mathbf{a})\}^{1/2} \tag{4}$$

where $\mathbf{a}$ and R are the center and the radius of the hypersphere, respectively. Note that δ_i^+ increases with growing ρ_i. Hence, to enclose the data point with increased δ_i^+ owing to a higher ρ_i, the radius of a minimum-sized hypersphere should be increased. The data point with higher relative density degree has stronger influence on the search of the minimum-sized hypersphere.

To incorporate the density degrees of negative data, we similarly define a *negative density-induced distance* δ_l^- between negative data $\mathbf{x}_l$ and $\mathbf{a}$, the center of the hyperspherical description of a target data set as

$$\delta_l^- \equiv \left\{\frac{1}{\rho_l}(\mathbf{x}_l - \mathbf{a}) \cdot (\mathbf{x}_l - \mathbf{a})\right\}^{1/2}. \tag{5}$$

Note that, contrary to δ_l^+, δ_l^- decreases with an increasing ρ_l. Hence, to exclude the negative data point with a decreased δ_l^- owing to a higher ρ_l, the radius of a compact hypersphere should be decreased; the negative data point with a higher relative density degree gives a higher penalty on the search of the compact hypersphere for a target data set.

B. Mathematical Formulation of D-SVDD

First, we find the optimal hypersphere that includes most target data by reflecting the relative density degrees of the target data. In this case, we use δ_i^+ and slack variable $\zeta_i(\geq 0)$ for the permission of training error for each target data point. Then, we can obtain the optimal hypersphere $(\mathbf{a}, R)$ by minimizing the objective function O

$$O = R^2 + C^+\sum_{i=1}^{n}\zeta_i \tag{6}$$

subject to $\rho_i(\mathbf{x}_i - \mathbf{a}) \cdot (\mathbf{x}_i - \mathbf{a}) \leq R^2 + \zeta_i$ where $C^+(> 0)$ is the control parameter in the C-SVDD [5]. Note that $\zeta_i = (\delta_i^+)^2 - R^2$ for training error data; otherwise, it is 0. It implies that ζ_i also contains the information of a relative density degree of $\mathbf{x}_i$.

Similar to C-SVDD [1], [5], the dual problem can be obtained by maximizing $D(\alpha)$ using Lagrange multipliers

$$D(\alpha) = \sum_{i=1}^{n}\alpha_i\rho_i\mathbf{x}_i \cdot \mathbf{x}_i - \frac{1}{T}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j\rho_i\rho_j\mathbf{x}_i \cdot \mathbf{x}_j \tag{7}$$

subject to $\sum_{i=1}^{n}\alpha_i = 1, 0 \leq \alpha_i \leq C^+$ and $T = \sum_{i=1}^{n}\alpha_i\rho_i$.

When negative data is available, our proposed method can also utilize them to improve the description of the target data set. In this case, using the negative density-induced distance and another slack variable $\zeta_l(\geq 0)$ for the possibility of training error in each negative data, we find the optimal hypersphere that includes most target data and excludes most negative data. Here, $\zeta_l = R^2 - (\delta_l^-)^2$ for negative training error data. Thus, the new objective function is defined as

$$O = R^2 + C^+\sum_{i=1}^{n}\zeta_i + C^-\sum_{l=1}^{m}\zeta_l \tag{8}$$

subject to $\rho_i(\mathbf{x}_i - \mathbf{a}) \cdot (\mathbf{x}_i - \mathbf{a}) \leq R^2 + \zeta_i$ and $(1/\rho_l)(\mathbf{x}_l - \mathbf{a}) \cdot (\mathbf{x}_l - \mathbf{a}) < R^2 - \zeta_l$, where m is the number of negative data and $C^- > 0$ is a control parameter similar to that of the C-SVDD [1], [5].

Similar to the previous case, the dual form of this case can be represented by maximizing $D(\alpha)$

$$D(\alpha) = \sum_{k=1}^{N}\alpha_k'\rho_k'\mathbf{x}_k \cdot \mathbf{x}_k - \frac{1}{T}\sum_{p=1}^{N}\sum_{q=1}^{N}\alpha_p'\alpha_q'\rho_p'\rho_q'\mathbf{x}_p \cdot \mathbf{x}_q \tag{9}$$

subject to $\sum_{k=1}^{N} y_k\alpha_k = 1, 0 \leq \alpha_i \leq C^+, 0 \leq \alpha_l \leq C^-, T = \sum_{k=1}^{N} y_k\alpha_k\rho_k'$, and $\alpha_k' = y_k\alpha_k$, where y_k is the label of $\mathbf{x}_k$ ($y_k = 1$ for a target data point; otherwise, $y_k = -1$) and N is the total number of data ($N = n + m$). Here, $\rho_k' = \rho_i$ for target data $\mathbf{x}_i$ and $\rho_k' = 1/\rho_l$ for negative data $\mathbf{x}_l$. Note that the dual form of this case retains the Lagrange multipliers α_i and omits the other variables and Lagrange multipliers β_i. Moreover, when $\rho_i = 1$ (and $\rho_l = 1$), this dual representation is equivalent to the formalism of a C-SVDD [5]; this means that the proposed method can act like the C-SVDD.

As seen in (7) and (9), the dual forms of the objective function of D-SVDD are represented entirely in terms of inner products of input vector pairs. Thus, we can kernelize D-SVDD to find a more flexible description of D-SVDD. The kernelized version of the dual representation in (9) is

$$D(\alpha) = \sum_{k=1}^{N}\alpha_k'\rho_k'K(\mathbf{x}_k, \mathbf{x}_k) - \frac{1}{T}\sum_{p=1}^{N}\sum_{q=1}^{N}\alpha_p'\alpha_q'\rho_p'\rho_q'K(\mathbf{x}_p, \mathbf{x}_q) \tag{10}$$

where $K(\cdot, \cdot)$ is a kernel function [9], [14], [15] and the constraints are the same with those of (9).

As shown in (7), (9), and (10), the dual forms are linearly constrained optimization problems. Thus, to solve these problems, we adapt the Powell's TOLMIN procedure [16], [17], which solves linearly constrained optimization problems by solving a sequence of quadratic programming subproblems to minimize the sum of constraint or bound violations. After solving the dual form in (10), the center $\mathbf{a}$ of solution can be calculated by

$$\mathbf{a} = \frac{\sum_{k=1}^{N}\alpha_k'\rho_k'\mathbf{x}_k}{T} \tag{11}$$

and the radius R is calculated by the δ_i^+ distance between $\mathbf{a}$ and any target data $\mathbf{x}_i$ of which $0 < \alpha_i' < C^+$. Note that, different from the C-SVDD [5], the center of the optimal hypersphere is weighted by the

TABLE I
AVERAGE ERROR RATES AND THOSE STANDARD DEVIATIONS (%) OF TEN INDEPENDENT RUNS FOR FIVE DATA SETS WHEN TRAINED USING ONLY TARGET DATA

Class	k-NNDD			Parzen-WD	C-SVDD			D-SVDD-I			D-SVDD-II		
	$k=1$	$k=3$	$k=5$	best	P3	P5	RBF	P3	P5	RBF	P3	P5	RBF
BREAST CANCER 0	26.94	33.40	36.55	4.62	5.89	5.94	5.09	5.12	**4.15**	**4.15**	5.24	5.24	4.51
	±2.4	±1.94	±2.66	±0.12	±0.74	±0.73	±0.63	±0.52	±0.64	±0.56	±0.44	±0.44	±0.36
1	30.29	26.57	25.84	15.05	4.69	4.74	4.71	3.93	4.15	**3.92**	4.21	4.13	4.31
	±2.23	±8.29	±7.62	±0.4	±0.45	±0.43	±0.51	±0.41	±0.38	±0.38	±0.36	±0.4	±0.41
Total	28.62	29.99	31.20	9.84	5.29	5.34	4.90	4.53	4.15	**4.03**	4.73	4.69	4.41
HEPATITIS 0	28.32	30.58	28.97	24.19	26.84	26.84	26.32	25.10	25.16	**20.52**	20.90	20.90	20.58
	±2.66	±2.49	±4.21	±1.33	±2.11	±3.9	±1.84	±1.06	±1.82	±2.13	±0.62	±0.62	±0.20
1	30.58	23.29	21.48	19.03	33.61	33.61	19.94	33.35	33.42	18.45	33.48	33.48	**17.71**
	±2.04	±3.25	±2.13	±1.15	±3.83	±2.08	±0.64	±1.52	±3.84	±1.44	±3.85	±3.84	±1.36
Total	29.45	26.94	25.23	21.61	30.23	30.23	23.13	29.23	29.29	19.48	27.19	27.19	**19.15**
IRIS 0	4.20	6.73	8.07	**0.00**	3.47	3.47	3.47	3.13	3.20	0.40	3.33	3.33	0.53
	±1.66	±1.87	±1.95	±1.08	±1.08	±1.08	±1.08	±1.08	±0.88	±1.08	±0.99	±0.99	±0.61
1	13.93	14.20	15.53	7.20	10.33	10.13	7.73	9.20	9.40	5.87	9.47	9.47	**5.53**
	±1.95	±3.82	±3.77	±0.78	±0.57	±0.71	±0.78	±0.76	±0.66	±0.53	±0.98	±0.76	±0.77
2	18.00	19.20	19.33	11.93	13.53	13.73	9.33	13.40	13.67	**8.07**	13.40	13.60	8.33
	±2.39	±4.2	±2.88	±2.04	±1.86	±1.74	±2.43	±2.21	±2.25	±1.86	±2.4	±2.29	±1.89
Total	12.04	13.38	14.31	6.38	9.11	9.11	6.84	8.58	8.76	**4.78**	8.73	8.80	4.80
LEUKEMIA 0	30.53	23.42	28.42	12.11	18.68	18.68	16.32	7.63	7.63	7.37	9.74	9.74	**7.11**
	±9.78	±6.38	±13.8	±4.99	±7.49	±7.49	±4.08	±5.75	±5.75	±5.79	±6.08	±6.08	±5.12
1	12.63	12.63	13.42	8.95	18.68	18.68	16.84	5.79	5.79	**4.74**	**4.74**	**4.74**	**4.74**
	±4.08	±3.68	±8.45	±3.09	±7.07	±7.07	±2.83	±1.66	±1.66	±1.36	±1.66	±1.66	±1.66
Total	21.58	18.03	20.92	10.53	18.68	18.68	16.58	6.71	6.71	6.05	7.24	7.24	**5.92**
WINE 0	7.08	9.78	12.70	2.08	5.51	5.62	5.51	1.91	1.91	1.63	1.74	1.80	**1.46**
	±1.68	±3.19	±4.91	±1.58	±1.58	±1.47	±1.58	±1.58	±0.6	±0.64	±0.67	±0.64	±0.67
1	28.54	29.16	26.18	19.94	25.51	25.45	19.72	25.45	25.34	**16.90**	25.51	25.45	17.08
	±2.89	±2.63	±4.34	±1.5	±2.54	±2.61	±1.14	±1.5	±2.5	±1.35	±2.54	±2.61	±1.03
2	7.42	12.25	11.40	4.94	6.01	6.01	5.84	4.55	4.55	3.76	4.61	4.61	**3.65**
	±2.28	±3.66	±2.34	±1.38	±1.27	±1.27	±1.36	±1.38	±1.07	±0.95	±0.98	±0.98	±0.96
Total	14.35	17.06	16.76	8.99	12.34	12.36	10.36	10.64	10.60	7.43	10.62	10.62	**7.40**

relative density degree ρ_k'; the center is shifted to a higher density region.

III. COMPARISON RESULTS

To investigate the success of these attempts and analyze the proposed method, we conducted various tests with five real data sets: the BREAST CANCER Wisconsin Database, the HEPATITIS Database, the IRIS Plant Database, the WINE Recognition Database from the University of California at Irvine KDD Archive [18], and the LEUKEMIA Database of Golub *et al.* [19]. After preparing the test data sets, we compared the performance of the proposed method with five other well-known methods: a k-nearest-neighbor data description method (k-NNDD) [5], a Parzen-window density method [5], a C-SVDD, a k-nearest neighbor classifier (k-NNC), and SVMs. The model parameters and other kernel parameters were found by cross validation to identify the best solutions of each method [5], [10].

The average error rates and those standard deviations of prediction accuracies of ten independent runs of twofold cross validations are given in Tables I and II. The label of a target data class is indicated in the first columns of these two tables and the data in other classes are the candidates of negative data. When only target data is used in training (Table I), for the BREAST CANCER data set, the three versions of the k-NNDD method showed 26.94%, 33.40%, and 36.55% as average error rates when the label of a target class is 0. For the same data sets, the Parzen-window density method showed 4.62% error rate; and the C-SVDD showed 5.89%, 5.94%, and 5.09% error, respectively. The proposed D-SVDD, however, showed 5.12%, 4.15%, and 4.15% error rates when Method-I of (2) was used, and showed 5.24%, 5.24%, and 4.51% error rates when Method-II of (3) was used. That was the C-SVDD comparable to the Parzen-window density method, and these two methods outperformed highly the k-NNDD method, and the proposed D-SVDDs outperformed slightly the C-SVDD in all versions used including the Parzen-window method, and the D-SVDDs with a Gaussian radial basis function (RBF) kernel function had the best performance (as indicated in bold type). Moreover, there was no big difference between the D-SVDD with the Method-I and D-SVDD with Method-II.

Similar results were obtained for other data sets, but the improvement of the D-SVDD over the C-SVDD was more conspicuous. Results on the IRIS data set, for example, the average error rate of the C-SVDD with a Gaussian RBF kernel function was 6.84% in all. On the contrary, the D-SVDDs showed 4.78% and 4.80% error rates according to the relative density extraction methods, respectively. Especially for the LEUKEMIA data set, the improvement was most prominent; the D-SVDD with Method-II obtained a 5.92% error rate, whereas the C-SVDD showed a 16.58% error rate when a Gaussian RBF kernel function was used.

When negative data was also used in training (Table II), similar phenomena with the previous case occurred, but the error rates of the proposed methods were highly decreased. In this case, the performance of the proposed methods were comparable with the SVMs and the k-NNC; the proposed method showed even better performance than the SVMs and the k-NNC for HEPATITIS, LEUKEMIA, and WINE data sets at the given test conditions. For the WINE data set, for example, the average error rate was 1.22% for the D-SVDD with Method-I, whereas for the C-SVDD, the error rate was 7.75%. The result was better than those of k-NNC (2.51%) and SVMs (1.40%). These kinds of improvement were more remarkable for the LEUKEMIA data set (Table II).

From Tables I and II, we conclude that the proposed method showed better prediction accuracies than the conventional data description methods including the C-SVDD for all of the tested cases, regardless

TABLE II
AVERAGE ERROR RATES AND THOSE STANDARD DEVIATIONS (%) OF TEN INDEPENDENT RUNS FOR FIVE DATA SETS WHEN TRAINED USING BOTH TARGET DATA AND NEGATIVE DATA

Class	C-SVDD			D-SVDD with M-I			D-SVDD with M-II			k-NN	SVMs			
	P3	P5	RBF	P3	P5	RBF	P3	P5	RBF	k=best	Lin	P3	P5	RBF
BREAST CANCER 0	4.98	4.99	4.81	4.23	4.23	**4.01**	4.36	4.36	4.51	3.38	3.39	4.13	6.37	**3.06**
	±0.49	±0.53	±0.46	±0.49	±0.51	±0.52	±0.35	±0.51	±0.35	±0.41	±0.15	±0.46	±0.76	±0.15
1	5.29	5.32	5.57	2.96	2.98	**3.15**	3.18	3.19	3.25	3.25	3.32	4.45	6.22	**2.99**
	±0.55	±0.53	±0.59	±0.22	±0.23	±0.41	±0.23	±0.1	±0.23	±0.29	±0.41	±0.66	±0.87	±0.21
Total	5.14	5.16	5.19	3.60	3.61	**3.58**	3.77	3.78	3.88	3.31	3.35	4.29	6.29	**3.03**
HEPATITIS 0	28.97	29.03	28.65	28.39	28.13	16.13	26.13	25.87	**15.23**	17.81	19.23	21.94	22.00	**17.03**
	±2.11	±4.01	±3.76	±3.76	±3.87	±1.21	±3.91	±3.74	±1.56	±1.86	±2.06	±3.91	±3.85	±1.98
1	19.48	19.48	18.13	16.39	16.39	**13.91**	16.19	16.13	14.52	19.48	19.35	22.26	21.55	**17.03**
	±3.82	±2.08	±1.27	±1.27	±1.46	±1.47	±1.5	±1.52	±1.56	±2.58	±0.96	±2.4	±1.62	±2.20
Total	24.23	24.26	23.39	22.39	22.26	15.02	21.16	21.00	**14.87**	18.65	19.29	22.10	21.77	**17.03**
IRIS 0	3.47	3.47	3.47	3.13	3.20	**0.20**	3.33	3.33	0.27	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
	±1.08	±1.08	±1.08	±1.04	±1.02	±0.32	±1.08	±1.08	±0.34	±0	±0	±0	±0	±0.00
1	9.07	8.93	7.47	8.27	8.00	**2.80**	8.13	8.13	**2.80**	4.33	25.80	4.87	7.40	**3.13**
	±0.72	±0.78	±0.76	±0.9	±1.06	±0.64	±0.72	±0.78	±0.64	±0.9	±0.95	±1.09	±1.19	±0.63
2	9.53	9.67	7.87	9.27	9.00	**3.60**	9.27	9.07	3.87	3.53	2.60	4.27	5.27	**2.53**
	±2.16	±2.04	±2.41	±2.09	±2.11	±0.65	±2.09	±2.11	±0.61	±0.89	±0.91	±1.45	±1.42	±0.69
Total	7.36	7.36	6.27	6.89	6.73	**2.20**	6.91	6.84	2.31	2.62	9.47	3.04	4.22	**1.89**
LEUKEMIA 0	18.68	18.68	13.95	3.68	3.68	**1.05**	3.95	3.95	**1.05**	3.95	**1.84**	5.00	8.68	**1.84**
	±7.49	±7.49	±4.31	±2.22	±2.22	±1.36	±2.24	±2.24	±1.36	±3.1	±2.5	±3.81	±3.73	±2.50
1	18.68	18.68	15.26	5.79	5.79	2.89	4.21	4.21	**2.63**	6.32	4.21	7.37	10.26	**3.95**
	±7.83	±7.07	±3.23	±2.42	±2.42	±2.48	±1.36	±1.36	±2.3	±3.55	±2.83	±5.52	±5.61	±2.84
Total	18.68	18.68	14.61	4.74	4.74	1.97	4.08	4.08	**1.84**	5.13	3.03	6.18	9.47	**2.89**
WINE 0	5.51	5.62	5.51	1.74	2.00	**0.22**	1.63	2.00	0.79	1.97	1.12	1.80	1.85	**0.11**
	±1.58	±1.47	±1.58	±1.32	±0.59	±0.4	±0.65	±0.65	±0.4	±0.55	±0.59	±0.83	±0.96	±0.24
1	12.42	12.42	11.97	9.61	11.47	1.91	10.17	12.07	**1.74**	3.82	2.70	4.27	5.17	**2.53**
	±1.49	±1.49	±1.5	±1.44	±1.09	±0.54	±1.46	±1.46	±0.47	±1.15	±0.83	±1.78	±2.4	±1.10
2	5.84	5.79	5.79	4.33	5.20	1.52	4.10	4.87	**1.40**	1.74	**1.35**	2.98	3.93	1.57
	±1.41	±1.43	±1.38	±0.75	±0.91	±0.85	±0.75	±0.75	±0.99	±0.67	±0.47	±1.3	±1.59	±0.69
Total	7.92	7.94	7.75	5.22	5.15	**1.22**	5.32	6.31	1.31	2.51	1.72	3.01	3.65	**1.40**

of the types of kernel functions or regardless of whether the negative data was used in training or not. Moreover, the best performance was obtained when the D-SVDD with Gaussian kernel functions were used. Furthermore, when negative data was used in training (Table II), the performance of the D-SVDD was comparable to those of the conventional well-known multiclass classification methods such as SVMs and k-NNC; despite, it is generally accepted that multiclass classification methods outperform data description methods [5]. That is because multiclass classification methods are invented to give the best separation without considering the volume of the data description [5].

IV. APPLICATION TO PROTEIN LOCALIZATION PREDICTION

Subcellular protein localization, the location where a protein resides within a cell, is one of the key functional characteristics of proteins [20], [21]. An automatic and efficient prediction method for the protein subcellular localization is highly required owing to the need for large-scale genome analysis [20]. From a machine learning point of view, a data set of protein localization has several characteristics: The data set has too many classes, it is a "multilabel" data set, and it is too "imbalanced" [21]. Even though many previous works have been invented for the prediction of protein subcellular localization, none of them have tackled such characteristics effectively [20].

We currently think that the proposed D-SVDD is one of good candidate methods for protein localization prediction. It is a one-class classification method, which is suitable for imbalanced data sets since it finds a compact description for a target data independently from other data [5]. Moreover, it is easily used for the data set whose number of classes is large owing to linear complexity with regard to the number of classes. However, basically the proposed D-SVDD is not for a multiclass and multilabel problem. For the protein localization problem, thus, we extend the proposed D-SVDD method by adopting the "one-versus-the other" approach as the following procedure.

1) If a training data set is given, we divide it by class into a target data set and a negative data set. For a label l_i, for instance, a data point whose label set has l_i is included in the target data; otherwise, it is included in the negative data set.
2) If a target data set and a negative data set are prepared for each class, we find the optimal boundary of the target data by using the cross-validation method.
3) We calculate the degree of membership for each class l_i for a test data point $\mathbf{x}_t$ using a scoring function

$$f(\mathbf{x}_t, l_i) = \frac{R_i}{d(\mathbf{x}_t, \mathbf{a}_i)} \tag{12}$$

where $(\mathbf{a}_i, R_i)$ is the optimal hypersphere of the target data that are included in the class label l_i. Note that this scoring function reports a higher value for a test data point with smaller Euclidean distance between $\mathbf{x}_t$ and $\mathbf{a}_i$ regarding distance of R_i.
4) Finally, according to the values of the scoring function for all classes, we rank the labels, and report them.

With this procedure, we can easily and intuitively extend the D-SVDD for a multiclass and multilabel classification problem like the protein localization prediction.

To evaluate the performance of the extended D-SVDD method, we represent a protein in three different ways using several famous ways [20], and make three data sets: Dataset-I, Dataset-II, and Dataset-III (see Table III). In Dataset-I, we used pair-AAC features, and gapped-AAC for sequence information including the amino acid composition (AAC)[20], [22]. In the Dataset-II, using 2372 unique motifs from InterPro Database, we represent a protein in a vector in a 2372-dimensional space [20]. For Dataset-III, we combine the

TABLE III
CHARACTERISTICS OF PROTEINS IN THE ORIGINAL HUH ET AL. DATA SET AND THREE TRAINING DATA SETS

Subcellular Localization	Huh *et al.* Dataset	Dataset-I	Dataset-II	Dataset-III
$\tilde{N}$	5184	5184	4032	4032
N	3914	3914	3017	3901
Dim. of Features		9620	2372	11992
Coverage		100%	77.08% ($\frac{3017}{3914}$)	77.08% ($\frac{3017}{3914}$)

TABLE IV
PREDICTION PERFORMANCE (%) OF ISORT AND EXTENDED D-SVDD TO THE THREE DATA SETS

Measure	Dataset-I		Dataset-II		Dataset-III	
	ISort	D-SVDD	ISort	D-SVDD	ISort	D-SVDD
Measure-I	65.14	73.89	69.94	82.40	75.90	83.49
Measure-II	35.91	53.09	44.27	56.32	49.16	57.24
Measure-III	10.02	19.10	15.33	44.61	20.25	46.50

previous two features only for the proteins that have more than one motif in the unique motif set.

For multilabel learning paradigms, only one measure is not sufficient to evaluate the performance of a predictor owing to the variety of correctness in prediction [23]. Thus, we use three measures (Measure-I, Measure-II, and Measure-III) for the evaluation of a protein localization predictor. First, to check the overall success rate regarding the total number of the unique proteins N, we define Measure-I as $(1/N)\sum_{i=1}^{N}\psi[L(P_i), Y_i^k]$, where N is the total number of different proteins, $L(P_i)$ is the true label set of a protein P_i, $Y_i{}^k$ is the predicted top-k labels by a predictor. Moreover, $\psi[L(P_i), Y_i^k] = 1$ if any label in $Y_i{}^k$ is in $L(P_i)$; otherwise, it is 0. We used $k = 3$ in this letter since the numbers of true localization sites of most proteins are less than or equal to 3 [21].

To check the overall success rate regarding the total number of classified proteins in each class $\tilde{N}$, we define Measure-II as $(1)/(\tilde{N})\sum_{i=1}^{N}\Psi[L(P_i), Y_i^{k_i}]$, where $\tilde{N}$ is the total number of classified proteins in each class, $L(P_i)$ is the true label set of a protein P_i, $Y_i{}^{k_i}$ is the predicted top-k_i labels by a predictor, and the $\Psi[\cdot,\cdot]$ function returns the correct number of labels which is predicted correctly.

Finally, to check the average rate of the success rates of each class, we define Measure-III as $(1/\mu)\sum_{l=1}^{\mu}((1)/(\widetilde{n}_l)\sum_{i=1}^{\widetilde{n}_l}\Delta[Y_i^{k_i}, l])$, where μ is a total number of classes, l is a label index, $\tilde{n}_l$ is the number of proteins in the lth label, $Y_i^{k_i}$ is the predicted top-k_i label of a protein P_i by a predictor. Here, $\Delta[Y_i^{k_i}, l] = 1$ if any label in $Y_i^{k_i}$ is equal to l; otherwise, it is 0.

For competitive analysis, we compared the performance of the proposed method with the ISort method [20] to the three data sets. Up to now, ISort method showed the best performance for the prediction of yeast protein multiple localization [20]. For the proposed method, we used a Gaussian RBF kernel function [10], and used Method-I for relative density degree extraction owing to a high dimensional feature space.

The results of the ISort method and the proposed method of a twofold cross validation for the three data sets are given in Table IV. From these results, we could conclude that the extended D-SVDD method outperformed the ISort method for all three data sets, regardless of the kind of evaluation measure. Moreover, motif information could increase the prediction accuracy of the two methods considered even though the coverage of motif information is lower than that of AAC-based information. Furthermore, the best performance was obtained when both features were used.

V. CONCLUSION

In this letter, we have proposed a novel method incorporating density distribution of a given target data set when identifying the hyperspherical data description of the data set. To reflect the density distribution, we associated each data point with a relative density degree, and proposed two kinds of density-induced distance measurement based on the degrees. Using the distance measurements, we developed a new SVDD method named D-SVDD. It was demonstrated that the proposed method outperformed the other data description methods including the conventional SVDD for all tested data sets, regardless of the kind of kernel function and regardless of the use of negative data in a training stage. When the information of negative data was available, the performance of the proposed method was also comparable to well-known multi class classifiers such as k-NNC and SVMs. Moreover, for a real application, we extended the proposed method to the protein localization prediction problem which is an imbalanced multiclass and multilabel problem, and observed promising results.

Currently, the time complexity of the proposed method is $O(kN^3)$ where N is the number of quadratic programming subproblems to solve linearly constrained optimization problems [16], [17]. Thus, to apply the proposed method to larger data sets, more research on reducing the computational time is required. Moreover, more formal justification of the proposed method is valuable for future work.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their helpful comments and Chung Moon Soul Center for Bioinformation and Bioelectronics, KAIST and the IBM Shared University Research (SUR) program for providing research and computing facilities.

REFERENCES

[1] D. M. J. Tax, "One-class classification: Concept-learning in the absence of counter-examples," Ph.D. dissertation, Electr. Eng., Math. Comp. Sci., Delft Univ. Technology, Delft, The Netherlands, Jun. 2001.
[2] C. Bishop, Mixture Density Networks Neural Computation Research Group, Aston Univ., Birmingham, U.K., Tech. Rep. NCRB 4288, 1994.
[3] G. Ritter and M. T. Gallegos, "Outliers in statistical pattern recognition and an application to automatic chromosome classification," *Pattern Recognit. Lett.*, vol. 18, pp. 525–539, 1997.
[4] E. Knorr, R. Ng, and V. Tucakov, "Distance-based outliers: Algorithms and applications," *VLDB J.*, vol. 8, pp. 237–253, 1999.
[5] D. M. J. Tax and R. P. W. Duin, "Support vector data description," *Mach. Learn.*, vol. 54, pp. 45–66, 2004.
[6] C. Bishop, *Neural Networks for Pattern Recognition*. London, U.K.: Oxford Univ. Press, 1995.
[7] M. Moya and D. Hush, "Network constraints and multi-objective optimization for one-class classification," *Neural Netw.*, vol. 9, no. 3, pp. 463–474, 1996.
[8] K. R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 181–201, 2001.
[9] B. Schölkopf and A. J. Smola, *Learning with Kernels*. Cambridge, MA: MIT Press, 2002.
[10] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.
[11] I. El-Naqa, Y. Yang, M. Wernik, N. Galatsanos, and R. Nishikawa, "A support vector machine approach for detection of microcalsifications," *IEEE Trans. Med. Imag.*, vol. 21, no. 12, pp. 1552–1563, Dec. 2002.
[12] A. Navia-Vázquez, D. Gutiérrez-González, E. Parrado-Hernández, and J. J. Navarro-Abellán, "Distributed support vector machines," *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 1091–1097, Jul. 2006.
[13] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Stat.*, vol. 33, pp. 1065–1076, 1962.
[14] K. Ikeda, "Effects of kernel function on Nu support vector machines in extreme cases," *IEEE Trans. Neural Netw.*, vol. 17, no. 1, pp. 1–9, Jan. 2006.

[15] S. W. Kim and B. J. Oommen, "On utilizing search methods to select subspace dimensions for kernel-based nonlinear subspace classifiers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 1, pp. 136–141, Jan. 2005.
[16] M. J. D. Powell, A tolerant algorithm for linearly constrained optimizations calculations Univ. Cambridge, Cambridge, U.K., DAMTP Rep. NA17, 1988.
[17] M. J. D. Powell, TOLMIN: A Fortran package for linearly constrained optimizations calculations Univ. Cambridge, Cambridge, U.K., DAMTP Report NA2, 1989.
[18] C. L. Blake and C. J. Merz, UCI Repository of Machine Learning Database (1998) [Online]. Available: http://www.ics.uci.edu/mlearn/ML-Repository.html/
[19] T. Golub, "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, pp. 531–537, 1999.
[20] K. C. Chou and Y. D. Cai, "Predicting protein localization in budding yeast," *Bioinf.*, vol. 21, pp. 944–950, 2005.
[21] W. K. Huh, J. V. Falvo, L. C. Gerke, A. S. Carroll, R. W. Howson, J. S. Weissman, and E. K. O'Shea, "Global analysis of protein localization in budding yeast," *Nature*, vol. 425, pp. 686–691, 2003.
[22] K. J. Park and M. Kanehisa, "Prediction of protein subcellular locations by support vector machines suing compositions of amino acid and amino acid paris," *Bioinf.*, vol. 19, pp. 1656–1663, 2003.
[23] F. A. Thabtah, P. Cowling, and Y. Peng, "MMAC: A new multi-class, multi-label associative classification approach," in *Proc. 4th IEEE Int. Conf. Data Mining*, 2004, pp. 217–224.

Rival-Model Penalized Self-Organizing Map

Yiu-ming Cheung and Lap-tak Law

***Abstract*—As a typical data visualization technique, self-organizing map (SOM) has been extensively applied to data clustering, image analysis, dimension reduction, and so forth. In a conventional adaptive SOM, it needs to choose an appropriate learning rate whose value is monotonically reduced over time to ensure the convergence of the map, meanwhile being kept large enough so that the map is able to gradually learn the data topology. Otherwise, the SOM's performance may seriously deteriorate. In general, it is nontrivial to choose an appropriate monotonically decreasing function for such a learning rate. In this letter, we therefore propose a novel rival-model penalized self-organizing map (RPSOM) learning algorithm that, for each input, adaptively chooses several rivals of the best-matching unit (BMU) and penalizes their associated models, i.e., those parametric real vectors with the same dimension as the input vectors, a little far away from the input. Compared to the existing methods, this RPSOM utilizes a constant learning rate to circumvent the awkward selection of a monotonically decreased function for the learning rate, but still reaches a robust result. The numerical experiments have shown the efficacy of our algorithm.**

***Index Terms*—Constant learning rate, rival-model penalized self-organizing map (RPSOM), self-organizing map (SOM).**

I. Introduction

Self-organizing map (SOM) [9] and its variants, e.g., see [5], [7], [8], and [17], are one of the popular data visualization techniques that provide a topological mapping from the input space to the output space. Typically, an SOM map possesses a regular one or two-dimensional (2-D) grid of nodes. Each node (also called *neurons* interchangeably) in the grid is associated with a parametric real vector called *model* or *weight* that has the same dimension as the input vectors. The task of SOM is to learn those models so that the similar high-dimensional input data are mapped into one-dimensional (1-D) or 2-D output space with the topology as unchanged as possible. That is, the similar data in the input space under a measurement are placed physically close to each other on the map. This topology-reserved feature of SOM leads it to the wide applications to data visualization [12], [6], data clustering [3], image analysis [10], data mining [13], and so forth.

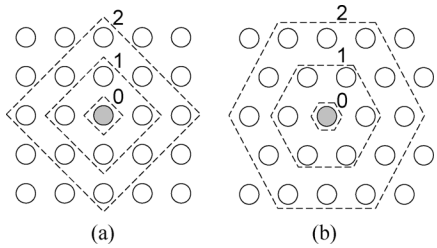

Fig. 1. Two neighborhood topologies commonly used in an SOM. (a) Rectangular neighborhood. (b) Hexagonal neighborhood. The number n next to the dashed lines denotes the n-neighborhood of the neuron in grey. The number of neurons in one-neighborhood in a k-polygonal neighborhood is k, i.e., the rectangular neighborhood has four one-neighborhood neurons while the hexagonal neighborhood has six one-neighborhood ones.

In general, a conventional adaptive SOM needs to initialize a learning rate and gradually reduces its value over time to ensure the convergence of the map. Usually, a small initial value of learning rate is prone to make the models stabilized at some locations of input space in an early training stage. As a result, the map is not well established. Hence, by rule of a thumb, the learning rate is often initialized at a relatively large value, and then gradually reduced over time using a monotonically decreasing function. If we reduce the learning rate very slowly, the map can learn the topology of inputs well with the small quantization error, but the map convergence needs a large number of iterations and becomes quite time-consuming. On the other hand, if we reduce the learning rate too quickly, the map will be likely trapped into a local suboptimal solution and finally led to the large quantization error. To the best of our knowledge, it is a nontrivial task to select an appropriate learning rate, in particular its associated monotonically decreasing function. In the literature, a two-phase training of the SOM has been further proposed by [14], which utilizes two learning rates to solve the previous selection problem in training the SOM. In the first phase, it keeps a large learning rate that aims at capturing the rough topological structure of the training data quickly. In general, the resulting map in the first phase is prone to error and the topological structure may not be well established. In the second phase, a much smaller learning rate is utilized to the trained map from the first phase, which aims at the fine-tuning topological map to ensure the map convergence. Nevertheless, the performance of the training algorithm is still sensitive to the time-varied learning rate.

In this letter, we therefore propose a new rival-model penalized self organizing map (RPSOM) learning algorithm inspired by the idea of the rival penalized competitive learning (RPCL) [15] and its recently improved variant, named rival penalization controlled competitive learning (RPCCL) approach [1], [2]. For each input, the RPSOM adaptively chooses several rivals of the best-matching unit (BMU) and penalizes their associated models a little far away from

Manuscript received September 5, 2004; revised June 20, 2006. This work was supported by the Research Grant Council of the Hong Kong SAR, China, under Project HKBU 2156/04E and Project HKBU 210306 and by the Faculty Research Grant of Hong Kong Baptist University, Hong Kong, under Project FRG/05–06/II-42.

The authors are with the Department of Computer Science, Hong Kong Baptist University, Hong Kong, SAR, China (e-mail: ymc@comp.hkbu.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2006.885039