Bio-Medical Materials and Engineering 26 (2015) S1813–S1820
DOI 10.3233/BME-151483
IOS Press

Improve the diagnosis of atrial hypertrophy with the local discriminative support vector machine[1]

Ping Ling[a,*], Dajin Gao[b], Xifeng Zhou[b], Zhining Huang[b] and Xiangsheng Rong[b]
[a]*College of Computer Science and Technology, Jiangsu Normal University, Xuzhou, 221116, China*
[b]*Department of Training, Air Force Logistics of P. L. A, Xuzhou 221000, China*

Abstract. Computer-aided diagnosis (CAD) approaches succeed in detecting a number of diseases, however, they are not good at addressing atrial hypertrophy disease due to the lack of training data. Support Vector Machine (SVM) is very popular in few CAD solutions to atrial hypertrophy. Yet the performance of SVM is moderate in atrial hypertrophy detection compared to its success in other classification problems. In this paper we propose a novel CAD algorithm, Local Discriminative SVM (LDSVM), to overcome the above-mentioned difficulty. LDSVM consists of a global SVM that is trained on the training data, and a local kNN that is trained based on the information of SVM and query. When a query arrives, SVM gives the initial decision. If the initial decision is less confident, local kNN begins to modify the initial decision. LDSVM improves the accuracy of SVM through some contributions: the risk tube, the discriminant information derived from SVM hyperplane, the new metric and the self-tuning size of neighborhood. Empirical evidence on real medical datasets show high performance of LDSVM over the peers in addressing atrial hypertrophy problems.

Keywords: Computer-aided diagnosis, support vector machine, discriminative direction, derivative of hyperplane function

1. Introduction

Computer-aided diagnosis (CAD) has always been one of the major research subjects in medical imaging and diagnostic radiology [1-4]. In computer science community, CAD approach belongs to pattern recognition or machine learning, with the aim to let computer yield a second opinion for medical staff. Although CAD approaches behave well in detecting many cardiovascular diseases [5-7], they are not good at addressing atrial hypertrophy disease. The reason is that the training data is extremely rare.

Neural network [8-10] and Support Vector Machine (SVM) [11, 12] are two of few CAD solutions to atrial hypertrophy problems. However, between the two tools, SVM performs better. On comparison with outstanding behaviors in other classification problems, performance of SVM is moderate for solving atrial hypertrophy problems. The underlying reasons for this lie in two aspects;

[1] This work is supported by the National Natural Science Foundation of China under Grant No. 61105129.
[*] Address for correspondence: Ping Ling, College of Computer Science and Technology, Jiangsu Normal University, Xuzhou, 221116, China. Tel./Fax: 86-516-83500307; E-mail: lingiceling@163.com.

firstly, the existing training data sets are of poor quality as there are errors during the data acquisition process. Further, the sharp difference in physiological information between individuals generates problems to find the accurate distribution of atrial data. Secondly, the gap between the normal and the abnormal is getting blurred, and the margin between two classes is getting vague. This hinders SVM to construct a qualified hyperplane within the margin.

In last decades, a number of SVM variants appeared: Eigenvalue Proximal Support Vector Machine [13], Twin Support Vector Machine [14], Multi-Weight Support Vector Machine [15], Least Square Twin Support Vector Machine [16], Least Square Projection Twin Support Vector Machine [17]; and some other variants [18-20]. These algorithms work well under many circumstances including computer-aided diagnosis. However, they only focus on common datasets and don’t care the characteristics of atrial hypertrophy data. Thus they perform moderately when they solve atrial hypertrophy problems. To overcome the above-mentioned difficulties, in this paper we propose a novel Local Discriminative Support Vector Machine (LDSVM). LDSVM consists of a global classifier, SVM, and a local classifier, k-nearest neighbor (kNN) [21, 22]. When a query arrives, the global SVM generates an initial decision. If the initial decision is less confident, then a local kNN is initiated to update the label decision. The novelty of LDSVM lies in the following; 1) Define a risk tube to observe whether the initial decision is confident enough or not. 2) Derive the discriminative information from SVM hyperplane. 3) A new metric is defined based on the derivative information. The neighborhood where kNN works is formulated with the new metric. Neighborhood size is decided by the query information. Experiments on real atrial hypertrophy datasets show LDSVM provides higher accuracy and appealing consumption compared with the peers.

In literatures, there are researchers who defined the rejection region for SVM. This rejection region is similar to the risk tube proposed in this paper. However existing research only identified the data falling into the rejection region, and didn’t do any further prediction to those hard queries. That actually does not do any help to upgrade the accuracy of SVM. Here LDSVM specifies the risk tube and handles the tube with a local-wise classifier, which improves the performance of the classifier.

2. Related work

For N samples: $(x_1, y_1)\ldots(x_N, y_N)$ sampling from $X \times Y$, X is the training dataset where $X = R^n$, and Y is the label set where $Y = \{1, -1\}$. SVM constructs the optimal hyperplane through optimizing:

$$\min \ \frac{1}{2}\Sigma_{i=1}^{N}\Sigma_{j=1}^{N} y_i y_j \alpha_i \alpha_j K(x_i, x_j) - \Sigma_{j=1}^{N}\alpha_j \quad \text{s.t.} \quad \Sigma_{i=1}^{N} y_i \alpha_i = 1\,, \quad 0 \le \alpha_i \le C \tag{1}$$

Therein K is the Kernel function. β_i is Lagrange multipliers. C is the penalty coefficient. Points with $0 < \beta_i < C$ are Support Vectors (SVs). Denote solution of (1) is β^*. The offset b of the hyperplane is:

$$b = y_j - \Sigma_{i=1}^{N} y_i \beta_i^* K(x_i, x_j) \tag{2}$$

Then the hyperplane is decided by:

$$f(x) = \Sigma_{i=1}^{N} \beta_i y_i K(x \cdot x_i) + b \tag{3}$$

kNN is a local classifier. When query arrives, kNN formulates a neighborhood for query and labels query as the most frequent class membership occurring among neighbors. Clearly kNN predicts query's membership according to the local information within the specified neighborhood.

3. The motivation and idea of LDSVM

SVM holds strong confidence when addressing the queries that are far apart from the hyperplane. However, it could lead to SVM's failure when handling queries near the hyperplane. This feature inspired us to improve the less-confident decisions of SVM with a local classifier that works well in the region where SVM is confused. In this paper we have chosen kNN to overcome this problem. In this approach a new metric is derived from the hyperplane information of SVM with an overall aim to discover the class boundaries. The hyperplane of SVM separates two classes. It means that along the direction of the hyperplane, data could be correctly classified. Consider the margin between two classes. In this region, it can be thought that the direction of the hyperplane can be replaced with the tangent direction of the hyperplane. In following description, the derivative of the hyperplane function is computed. This derivative information quantifies the tangent direction, so as to simulate the most discriminant direction.

In a word, LDSVM corresponds to the following steps.1) Preprocess the atrial hypertrophy data. 2) Construct SVM on training data; obtain the hyperplane function: $f(x)$. 3) For query Q, label it with SVM. 4) If Q falls in the risk tube, develop the new metric $\|\cdot\|_Q$. 5) Compute the neighborhood size, *NS*. 6) Formulate the neighborhood of Q and start local kNN classifier to update the label of Q.

4. Implementation of LDSVM

4.1. Tube specification

SVM labels queries according to which side query falls. The queries that fall closer to the hyperplane, have higher chances that they are classified by mistake. Here around the hyperplane a banding-like region is specified as the risk tube. If query falls in this region, the initial decision of SVM will be modified. In details, we think if the query satisfies the condition shown below then it falls in the risk tube:

$$|f(Q)| < 0.5 \tag{4}$$

Therein $f(x)$ is the hyperplane function of SVM. $|f(Q)|$ provides the distance of Q from the hyperplane. In the 2-dimensional data space, the risk tube generated by the above condition is illustrated in Figure 1.

4.2. Metric definition

The direction of hyperplane is described by the derivative of $f(x)$: $f'(x)$. Here it takes the individuality of queries into consideration and compute the derivative with respect to Q. There are two scenarios. If Q is on the hyperplane, say, $f(Q) = 0$, the gradient vector $f'(Q)$ can be utilized directly. If Q is not on the hyperplane, it should look for Q's nearest neighbor that is on the hyperplane. This

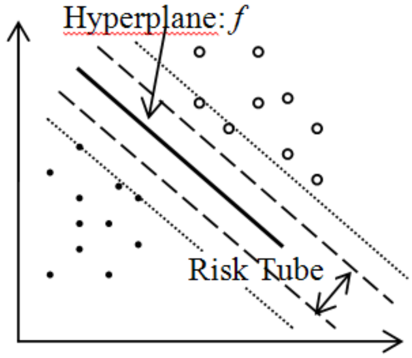

Fig. 1. The illustration of risk tube.

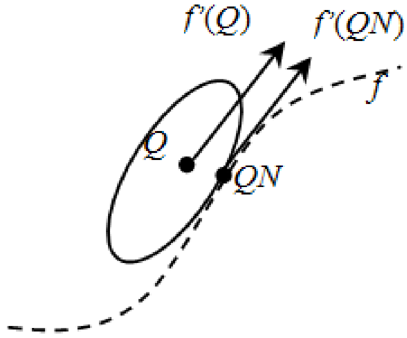

Fig. 2. f'(Q) and f'(QN).

neighbor is denoted as QN, and then QN can be probed via optimization as shown below, by optimization problem:

$$\min_{QN} \| Q - QN \| \qquad \text{s.t.} \quad f(QN) = 0 \tag{5}$$

Extra cost is incurred during optimization. If the local kNN is initiated, it implies Q is located within the risk tube. That is, Q is near the hyperplane. As shown in Figure 2 the direction expressed by $f'(Q)$ is very close to the direction expressed by $f'(QN)$. Therefore we simulate QN with Q.

Now in both the scenarios, $f'(Q)$ is used to define the new metric. Denote $G = f'(Q) = (G_1, G_2, \ldots, G_n)$, where G_i is the i^{th} element of $f'(Q)$. G_i tells the importance of the i^{th} dimension when describing this discriminant direction. In this paper the derivative information is used to weight data original coordinates. In details, for the i^{th} dimension, the weight coefficient is defined as:

$$\mu_i = \frac{\exp\left(\frac{|G_i|}{\Sigma_{j=1}^{n} |G_j|}\right)}{|f(Q)|} \tag{6}$$

Exponential mechanism avoids the dramatic changes in computation. Given x and y being any two points, the new metric that is customized to Q is defined as:

$$\| x - y \|_Q = \sqrt{\Sigma_{i=1}^{n} \ \mu_i (x_i - y_i)^2} \tag{7}$$

4.3. Neighborhood formulation

With the new metrics in hand, we can specify the size of neighborhood as follows. 1) Compute the distance between each training data, x, and the query: $\|x - Q\|_Q$. 2) Sort training data into the ascending order: $\{ x_{(i)} \mid \|x_{(i)} - Q\|_Q > \|x_{(i-1)} - Q\|_Q, i = 2 \ldots N \}$. 3) Find the maximum gap position in the list:

$$gap(Q) = \max_i \ \{ \| x_{(i)} - Q \|_Q - \| x_{(i-1)} - Q \|_Q \} \tag{8}$$

4) Let the neighborhood size $NS = gap(Q)$ and specify the nearest NS data $\{x_{(1)}, \ldots x_{(NS)}\}$ as the

neighbors of *Q*. *gap*(*Q*) implies the inherent boundary of a small region centered at *Q*. It is believed that data within that region are densely gathered. We view that region as the neighborhood of *Q*.

5. Application to atrial hypertrophy data

5.1. Data preprocessing

MGH/MF dataset and Fantasia dataset [23, 24] are considered as experimental datasets. They are downloaded, observed and processed by *WFDB* software [25]. MGH/MF includes the electrocardio-records of 250 patients; Fantasia dataset includes the electrocardio-records of 40 healthy persons. If one heartbeat has original 300 sampling points, we compute the sampling interval: 300/50 = 6. Then reserve the 6th sampling point and abandon from 1st to 5th sampling points; reserve the 12th sampling point and abandon from 7th to 11th sampling points, and so on; until 50 reserved sampling points are obtained. Therefore, a 50-dimensional vector describes one heartbeat. It should be noted here that its voltage describes each sampling point. Therefore, this 50-dimension vector records 50 voltages. In this paper, entries of such a 50-dimension vector are normalized by subtracting the mean voltage of corresponding dimension, with aim to ranging the electrocardio-data valued around zero. In experiments, some patient records and some healthy person records are selected randomly from MGH/MF and Fantasia to formulate four training and four testing datasets, which are named as a), b) c) and d), as shown in Table 1. Here '#' stands for the number of samples.

For LDSVM, its global SVM is equipped with Gaussian Kernel. The scale parameter and the penalty coefficient are tuned by 10-fold cross validation [26]. Neural network (NN) and classical SVM are conducted for comparison. Here NN uses a three-layer network. The number of neurons and the transfer function are reported in Table 2. The maximum iteration times of NN is set as 10^4. Classical SVM also employs Gaussian Kernel. Similarly, the scale parameter of Gaussian Kernel and the penalty coefficient of classical SVM are also tuned by 10-fold cross validation. To investigate the quality of the proposed metric, another version of LDSVM, named as $LDSVM_{EU}$, is conducted. $LDSVM_{EU}$ consists of global SVM and kNN that is based on Euclidean metric. That is, the neighborhood where kNN works is developed based on Euclidean metric.

Table 1

The specification of experimental datasets

	Training dataset		Testing dataset	
	# patients	# healthy persons	# patients	# healthy persons
a)	30	31	10	9
b)	38	37	10	3
c)	50	35	10	5
d)	65	35	10	5

Table 2

Details of NN

	# neurons	Transfer function
Input Layer	50	$g(x) = x$
Hidden Layer	10	$g(x) = \text{tansig}(x)$
Output Layer	1	$g(x) = \text{purelin}(x)$

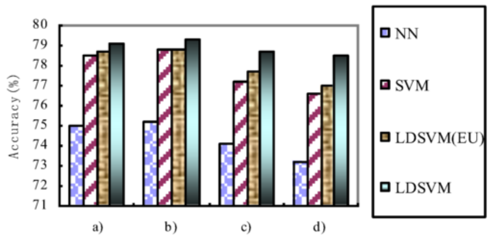

Fig. 3. Comparison of accuracy on four datasets.

Table 3

Theoretical time complexity analysis

	Time Complexity
NN	Unexpected
SVM	$O(N^3)$
LDSVM	$O(N^3) + O(N\log N)$

5.2. Experiment results

Experimental results are listed in Figure 3. From Figure 3, it is safe to conclude the following. 1) LDSVM presents the optimal results. $LDSVM_{EU}$ produces the second optimal results, followed by SVM. NN performs the worst. The advantage of LDSVM and $LDSVM_{EU}$ over classical SVM is attributed to the employment of local classifier to modify the initial decision. In the risk tube, the local kNN does effectively refine less-confident decisions. The advantage of LDSVM over $LDSVM_{EU}$ can be explained that the new metric is well done. The neighborhood formulated by the new metric can cover the neighbors with same class membership as the query. That greatly assists kNN to improve the unpleasant initial decisions. These experimental results verify the quality of the new metric and show the discriminant information derived from SVM hyperplane function can benefit the consequent label assignment. 2) In datasets a) and b), the sizes of two classes are very close. They are viewed as the balanced cases. In datasets c) and d), there is an obvious gap in the sizes of two classes. They are viewed as the imbalanced cases. According to experiment results, four classifiers do better jobs in the balanced cases in comparison to the imbalanced cases. For LDSVM, $LDSVM_{EU}$, and SVM, they construct the hyperplane around the central line sites of the margin between two classes. The imbalance between two sides would attract the hyperplane from the central line to the sideling sites. That inevitably affects the classification accuracy. For NN, the imbalanced classes affect updating of weight coefficients of network, and the final output.

5.3. Time complexity analysis

This section further explores the time consumption of NN, SVM and LDSVM. Table 3 lists the theoretical analysis, where N is the size of training dataset.

Therein the cost of NN is unexpected and the implementation depends on many factors, such as, the structure of network (including the number of layers and neurons), the termination condition (using the pre-specified iteration times or the threshold of error), etc. SVM solves a quadratic optimization problem, with the cost being $O(N^3)$. The time consumption of LDSVM consists of cost of global SVM and cost of local kNN. Global SVM's time complexity is $O(N^3)$. The time complexity of local kNN lies in sorting training data. We chose the most commonly used sorting algorithm, quick sorting algorithm to fulfill the sorting task. On average, quick sorting algorithm fulfills log*N* partitions, and in each partition, it does *N* comparisons. Hence, the time complexity of local kNN is O(*N*log*N*). LDSVM consumes more time than SVM. But in the era when computer hardware is in mushroom growth, the difference in cost between LDSVM and SVM will gradually reduce.

From the above experiments, it is evident that among existing CAD solutions to detect atrial hypertrophy diseases, LDSVM behaves the best; and in cost LDSVM is more appealing than others.

6. Conclusion

This paper proposes a Local Discriminative Support Vector Machine (LDSVM) to address atrial hypertrophy problems. LDSVM employs the kNN classifier to refine the less-confident decision of SVM. For that, a risk tube is defined. The queries falling into the tube are re-labeled. A new metric is derived from the hyperplane function of SVM. For local kNN, the neighborhood is developed based on the new metric and the neighborhood size is decided query-adaptively.

In the future, some medical diagnosis problems will utilize LDSVM. For example, in the prediction of disease-causing SNP, existing approaches are used to including sequences of protein paralogs, which affects the prediction accuracy. Wong, et al. [27] proposed an algorithm to fulfill the prediction using orthologous protein sequences to improve the prediction accuracy. With the scarcity of orthologous protein sequences, it is interesting to introduce LDSVM into such a problem, since LDSVM is equipped with local kNN to address the difficulty of the small-sized training data.

References

[1] T. Hothorn and B. Lausen, Bagging tree classifiers for laser scanning images: A data- and simulation-based strategy, Artificial Intelligence in Medicine **27** (2003), 65–79.

[2] A. Sharkey, N. Sharkey and S. Cross, Adapting an ensemble approach for the diagnosis of breast cancer, Proceedings of the 6th International Conference of Artificial Neural Network, Skovde, Sweden, 1998, pp. 281–286.

[3] Z.H. Zhou, Y. Jiang, Y.-B. Yang and S.F. Chen, Lung cancer cell identification based on artificial neural network ensembles, Artificial Intelligence in Medicine **24** (2002), 25–36.

[4] M. Li and Z.H. Zhou, Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples, IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans **37** (2007), 1088–1098.

[5] J.J. Shiraishi, Q. Li, D. Appelbaum and K. Doi, Computer-aided diagnosis and artificial intelligence in clinical imaging, Seminars in Nuclear Medicine **41** (2011), 449–462.

[6] A. El-Baz, G.M. Beache, G. Gimel'farb, K. Suzuki, K. Okada, A. Elnakib, A. Soliman and B. Abdollahi, Computer-aided diagnosis systems for lung cancer: Challenges and methodologies, International Journal of Biomedical Imaging **2013** (2013), 1–46.

[7] W.Y. Chang, A. Huang, C.Y. Yang, C.H. Lee, Y.C. Chen, T.Y. Wu and G.S. Chen, Computer-aided diagnosis of skin lesions using conventional digital photography: A reliability and feasibility study, Plos One **8** (2013), 1–9.

[8] S. Grossberg, Nonlinear neural networks: Principles, mechanisms, and architectures, Neural Networks **1** (1988), 17–61.

[9] A. Martin and P.L. Bartlett, Learning in Neural Networks: Theoretical Foundations, Cambridge University Press, UK, 1999.

[10] T.R. Fabián, J.R.S. Alberto, J.G.F. Francisco and B.T. Ivonne, A robust neuro-fuzzy classifier for the detection of cardiomegaly in digital chest radiographies, DYNA **81** (2014), 35–41.

[11] V. Vapnik, Statistical Learning Theory, Wiley New York, USA, 1998.

[12] S. Ingo, C, Andreas, J. Michael, K. Jon and B. Schölkopf, Support Vector Machines, Springer, New York, USA, 2008.

[13] O.L. Mangasarian and E.W. Wild, Multisurface proximal support vector classification via generalized eigenvalues, IEEE Transactions on Pattern Analysis and Machine Intelligence **28** (2006), 69–74.

[14] Jayadeva, R. Khemchandani and S. Chandra, Twin support vector machines for pattern classification, IEEE Transactions on Pattern Analysis and Machine Intelligence **29** (2007), 905–910.

[15] Q. Ye, C. Zhao, N. Ye and Y. Chen, Multi-weight vector projection support vector machines, Pattern Recognition Letters **31** (2010), 2006–2011.
[16] M.A. Kumar and M. Gopal, Least squares twin support vector machines for pattern classification, Expert Systems with Applications **36** (2009), 7535–7543.
[17] Y.H. Shao, N.Y. Deng and Z.M. Yang, Least squares recursive projection twin support vector machine for classification, Pattern Recognition **45** (2012), 2299–2307.
[18] Y. Ji and S.L. Sun, Multitask multiclass support vector machines: Model and experiments, Pattern Recognition **46** (2013), 914–924.
[19] X.J. Xie and S.L. Sun, Multitask centroid twin support vector machines, Neurocomputing **149** (2015), 1085–1091.
[20] X.J. Xie and S.L. Sun, Multi-view Laplacian twin support vector machines, Applied Intelligence **41** (2014), 1059–1068.
[21] K. Fukunage and P.M. Narendra, A branch and bound algorithm for computing k-nearest neighbors, IEEE Transactions on Computers **24** (1975), 750–753.
[22] L. Li, H.T. Liu, J. He, N.T. Zhang and W.T. Song, An improved KNN text classification algorithm based on density, IEEE International Conference on Cloud Computing and Intelligence Systems, Beijing, China, 2011, pp. 113–117.
[23] N. Iyengar, C.-K. Peng, R. Morin, A.L. Goldberger and L.A. Lipsitz, Age-related alterations in the fractal scaling of cardiac interbeat interval dynamics, American Journal of Physiology **271** (1996), 1078–1084.
[24] A.L. Goldberger, Amaral Lan, L. Glass, J.M. Hausdorff, PCh Ivanov, R.G. Mark, J.E. Mietus, G.B. Moody, C.-K. Peng and H.E. Stanley, PhysioNet: Components of a new research resource for complex physiologic signals, Circulation **101** (2000), e215–e220.
[25] G.B. Moody, WFDB applications guide, Harvard-MIT Divi Sion of Health Sciences and Technology **10** (2003), 1-20.
[26] S. Arlot and A. Celisse, A survey of cross-validation procedures for model selection, Statistics Survey **4** (2010), 40–79.
[27] K.-C. Wong and Z.L. Zhang, SNPdryad: Predicting deleterious non-synonymous human SNPs using only orthologous protein sequences, Bioinformatics **30** (2014), 1112–1119.