

Chapter 2

Classifier Selection Based on Support Vector Technique

Dajin Gao, Xiang-Sheng Rong, Xiang-Yang You, Ming Xu
and Fujiang Huo

Abstract This paper presents an ensemble approach consisting of global SVM and local SVM. Global SVM is estimated according to its decision confidence. Local SVM handles the query whose global decision is of low confidence. Local SVM is constructed over query's neighborhood, which is developed under the guidance of an informative metric. And its training is based on a query-based objective function. Global SVM helps to define the new metric. Some heuristics proposed to specify neighborhood size and hyper parameters. We present experimental evidence of classification performance improved by our schema over state of the arts on real datasets.

Keywords Local classifier · Global classifier · Support vector technique · New metric

2.1 Introduction

As a qualified classification algorithm, SVM [1] has proved efficiency in a wide range of applications from pattern recognition to function regression, and time series prediction. Its basic idea of structural risk minimization [2] equips SVM with high generalization. In spite of the success in most cases, SVM can't provide qualified decision for some outliers or some ambiguous data that are located

D. Gao (✉) · X.-S. Rong · X.-Y. You
Training Department, Xuzhou Air Force College of P. L. A, Xuzhou 221000, China
e-mail: rxs12@126.com

M. Xu · F. Huo
Department of Logistic Command, Xuzhou Air Force College of P. L. A,
Xuzhou 221000, China

around the decision interface. The reason lies in the generalization property of SVM to make it hold an overly high confidence in decisions on all inputs. However that space is not necessarily suitable everywhere, just like a global model is not reared to some local regions. A case of point is the point with $0 < f(x) < \delta$, assuming δ being a small value. Such a point takes much possibility to change its membership if being perturbed with small amount.

Essentially local SVM is in an adaptive and informative space, where the discriminate direction concerned with the query can be revealed. Global SVM helps to modify input space into new space over query's neighborhood, which is achieved by specifying hyper parameters of local space. The local property of new SVM is reflected in the objective function of local SVM. It adopts Kernel affinities as penalty coefficients of slack variables. Another point is the heuristic rules for global SVM hyper parameters which bring computation ease. The proposed approach is experimented on real datasets to find it does a better or competitive job than traditional SVM-based schemas and gives very competitive results compared with the state-of-the-art methods while using less computation cost.

2.2 Related Work

Three techniques used in this paper are given in brief. The first one is SVC. Let $x_i \in X$, $X = \mathbb{R}^n$ be the input space, and Φ be the nonlinear transformation from X to the feature space. To find a minimum hyper sphere that encloses all data, the optimal objective function with slack variable ξ_i is designed as:

$$\min_{R, \xi} \quad R^2 + C \sum_i \xi_i \text{ s.t. } \|\Phi(x_i) - a\|^2 \leq R^2 + \xi_i, \xi_i \geq 0 \quad (2.1)$$

a and R are the center and radius of sphere, C is the penalty parameter. Transfer its Lagrangian function into Wolf dual, and introduce Kernel trick, leading to:

$$\max_{\beta} \quad \sum_i \beta_i K(x_i, x_i) - \sum_{i,j} \beta_i \beta_j K(x_i, x_j) \text{ s.t. } \sum_i \beta_i = 1, 0 \leq \beta_i \leq C \quad (2.2)$$

Gaussian Kernel $K(x_i, x_j) = \exp(-q\|x_i - x_j\|^2)$ is used. Points with $\xi_i = 0$ and $0 < \xi_i < C$ are referred as non-bounded Support Vector (nbSV) and they describe cluster contours. Points with $\xi_i > 0$ and $\beta_i = C$ are bounded Support Vector (bSV).

k NN is a simple but attractive method. It labels the query as the most frequent class of neighborhood.

SA procedure [3, 4] is the third technique. SA obtains data spectral projections by eigen-decomposing a pairwise matrix H , which is usually the normalized affinity matrix. Select top p eigenvectors and form spectral embedding matrix S by stacking p eigenvectors in columns. Rows of S are data's spectral coordinates. Then it clusters spectrums with a simple method, and assigns point the same label as its spectrum.

2.3 NSVC Algorithm

NSVC uses the objective function of traditional SVC, but modifies its Kernel scale data-dependently so that data representatives (DRs) are extracted. Then SA is conducted on DRs to collect label information. Simultaneously, a new metric is defined, and this metric helps find query's NEI. That NEI is divided into sub neighborhoods (sNEI) in terms of classes. Each sNEI is enriched with convex hull technique. Label assignment is done in sNEI. The steps are:

1. Optimize: $\min R^2 + C \sum_i \xi_i$, to produce $\{\text{DRs}\}$.
2. Conduct SA on $\{\text{DRs}\}$, to obtain labels and new distance definition $\|\cdot\|_{\text{sd}}$.
3. For query $Q \neq \text{SV}$
4. Generate Q 's NEI according to $\|\cdot\|_{\text{sd}}$;
5. Divide NEI into sub region sNEI_j for involved class j .
6. $\text{esNEI}_j = \text{convex_hull}(\text{sNEI}_j)$.
7. Formulate weight w_j based on esNEI_j .
8. $t_j = \text{frequency of class } j \text{ in NEI}$.
9. Label (Q) = $\max_j \{t_j \cdot w_j\}$.

Here, the affinity matrix H of SA is normalized in the fashion $H' = D^{-1/2} H D^{-1/2}$, where D is diagonal-shape with $D_{ii} = \sum_{j=1}^n H_{ij}$. p controls the number of selected eigenvectors, and it is ups to the max gap in the descending eigenvalue list [5]. K-means method is used to classify data spectrum coordinates.

2.3.1 Self-Tuning Kernel Scale

This paper investigates q in data local context. For point x we set its scale factor as: $\sigma_x = \|x - x_r\|$. To measure affinity between x and y , their scale factors are combined together to develop $q = 1/\sigma^2 = \sigma_x \cdot \sigma_y$. This leads to the tuning Kernel:

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{\|x - x_r\| \cdot \|y - y_r\|}\right) \quad (2.3)$$

r is regarded as the size of the neighborhood. It is set as following steps: (a) Sort rows of Euclidean distance matrix $d(i, j)$ in an ascending order. (b) Let $\text{gap}(i) = \max_j \{d(i, j) - d(i, j-1)\}$. (c) $r = \text{average} \{\text{gap}(i)\}$.

Below is visual proof. For dataset nvSVs of tuning SVC. Clearly SVs are located on cluster contours and important positions where sharp changes of distribution density happen. They provide a sketch of dataset, and their NEIs are believed to cover the entire dataset. This justifies the feasibility of NSVC's labeling method.

2.3.2 Individual Setting Penalty Coefficient

In this paper C is parameterized by integrating diverse C_i values expected by points individually.

Set N as dataset size, and rows of the Kernel matrix $k(i, j)$ has been sorted in a descending order. With $Kgap(i) = \max_j \{k(i, j) - k(i, j + 1)\}$, the definition of C_i for x_i is:

$$C_i = \exp\left(-\frac{aveIn(i)/aveAll(i) - 1}{1 - Kgap(i)/N}\right) \quad (2.4)$$

Here,

$$aveIn(i) = \left(\sum_{j=1}^{Kgap(i)} k(i, j)\right) / Kgap(i) \quad (2.5)$$

$$aveAll(i) = \left(\sum_{j=1}^N k(i, j)\right) / N \quad (2.6)$$

$Kgap(i)$ acts as the size estimate of neighborhood. $aveIn(i)$ is the average affinity of x_i neighborhood. $aveAll(i)$ is the average affinity of x_i to all other points. Both of them tell density information. C_i reflects x_i 's individual demand to penalty term. So the global C is defined as the average of all individual C_i :

$$C = average\{C_i\} \quad (2.7)$$

2.4 Local SVM

When global SVM yields output of low confident, a query-based SVM is trained in query's neighborhood. That neighborhood is developed by a locally adaptive and informative metric, which is described next.

2.4.1 New Metric

Decision function of global SVM helps to derive new metric. Viewed under the light of theory, SVM decision function is optimal in the sense of structural risk minimization and therefore it is very desirable for seeking discriminates directions between classes. Viewed from geometry light, to any point x on level curve $f(x) = 0$, the gradient vector $f'(x)$ reveals the perpendicular orientation along which data can be well separated over x 's neighborhood. Integrate that orientation into metric definition and classification can be benefited a lot.

Without loss of generality, given query Q , we find its nearest neighbor P on the interface $f(x) = 0$ by:

$$\min_P \|Q - P\| \text{ s.t. } f(P) = 0. \quad (2.8)$$

It is regarded that discriminates information is rich over P 's neighborhood. But this optimization asks for extra cost, so we simulate P with Q and use $f'(Q)$ as the guidance to formulate new metric. Let $G_Q = f'(Q) = (G_{Q,1} \dots G_{Q,n})$, then the new distance is defined as

$$\|x - y\|_{new} = \sqrt{(x - y)^T \mu (x - y)}. \text{ With } \mu_i = \frac{\exp(B \cdot |G_{Q,i}|)}{\sum_{j=1}^n \exp(B \cdot |G_{Q,j}|)} \quad (2.9)$$

Therein, the exponential mechanism is added to guarantee value stability. B controls the influence of elements G_Q on the whole weights. It is set as $B = 1/|f(Q)|$. The nearer Q is to $f(x)$, the more effect of G_Q should be strengthened, and the more informative local metric is. On the contrary, if Q is far to $f(x)$, B gets to zero and weight $\mu_i = 1/n$, which results to the original distance definition.

In M -classification, we create 1-vs-r SVMs, so M decision functions f_j and $f'_j(Q)$ are involved. These gradient vectors are combined in a weighted fashion. We design combination weights proportion to decision function values in the sense that the closer Q is to f_j , more influence $f'_j(Q)$ makes in formulating the final discriminant orientation. Let $G_Q^j = f'_j(Q)$, so the comprehensive orientation is defined:

$$G_Q = \sum_{j=1}^M \left(1 - \frac{f_j(Q)}{\sum_{l=1}^M f_l(Q)}\right) \cdot G_Q^j \quad (2.10)$$

In the new feature space spanned by $\|\cdot\|_{new}$, the Kernel is updated into:

$$k_{new}(x, y) = \exp(-\|x - y\|_{new}^2) \quad (2.11)$$

2.5 Experiment Results

2.5.1 Test New Metric

Firstly the quality of new metric is checked by introducing it into the k NN procedure to develop a classifier that probes query's *NEI* based on *Sd* and labels query with weighted voting strategy. That classifier is named $WkNN$. Six datasets are taken from UCI Machine Learning Repository. In Table 2.2, $WkNN$ are compared on the average of 20 runs with following classifiers: (1) kNN . (2) SVMs of 1-vs-r version (SVM_{1r}). (3) SVMs of 1-vs-1 version (SVM_{11}) (4) C4.5 decision tree. (5) Machete [6]. (6) Scythe. (7) DANN. (8) Adamenn. Here, 30% data are sampled randomly for training.

Generally speaking, Adamenn achieves better performance than other classifiers by giving 4 optimal results of 6 experiments. It collects entire statistics about data distribution to define a globally informative metric, so its work is steady and fine. But its good behaviors are at the price of expensive consumption that is spent on tuning six parameters. WkNN achieves the optimal result in Banana, and follows Adamenn in other cases with a gentle distance to the optimal result. That demonstrates the validation of WkNN idea and the fine performance of this algorithm. If consider the computation ease brought by the self parameterization, WkNN is a more appealing choice in practice. DANN is on the second place. The metric employed by DANN approximates the weighted Chi-squared distance, which causes it fails in datasets of non-Gaussian distribution. Machete and Scythe are rooted from the same spirit, but the latter modifies the greedy nature of the former, so it improves the clustering accuracy in most cases.

Then it proceeds to SVM_{1r} and SVM_{1l}. They don't depend on the deriving new metric, but on constructing the wise separating surface. They work well in the scenarios where classes are non-linear separated by mapping data into a feature space and then transferring the non-linear classification into the linear classification. Here, six datasets involved are mostly linear-separated, but with irregular cluster shapes. That data distribution permits their performance, so they can't play all potential power. C4.5 and kNN work poorly due to their greedy idea and unsupervised partition respectively.

2.5.2 Test NSVC

Now NSVC is performed and compared with some popular clustering algorithms: K-means; traditional SVC; Girolami method; and NJW. For each algorithm, the minimum number of incorrectly clustered points is documented. And we also present results of another NSVC version that is encoded with a width searching approach. We find the best clustering result by running over a specified range of $1/\sigma^2$. This method is named as search-NSVC. As to Wine data, the fact that 178 points cover 13 dimensions leads to the wide spreading information and the weak neighborhood information in the local context. So the tuning approach exhibits little help to refine affinities and NSVC produces a high error rate.

Between two versions of NSVC, surely search-NSVC is better than the tuning version. The difference between two versions is not large. Among other four methods, NJW does the best job. Search-NSVC is competitive with it, which verifies the capacity of NSVC algorithm idea. Girolami's performance follows search-NSVC, then SVC and K-means in turn. Their work has apparent gap with the above three methods. Girolami sometimes are affected by the unsteady optimization process. SVC's challenges lie in its expensive labeling process, whose randomness degrades SVC's final results. But the non-linear map hidden by Kernel makes SVC does better than K-means, the method depends only on input space

Table 2.1 Comparisons on classification error (%)

Data	Diabetes	Ionosphere	Banana	Liver	Sonar	Waveform
k NN	1514.7	5.92	13.6	31.5	12.65	18.4
SVM _{lr}	8.12	6.72	14.3	28.4	11.12	18.0
SVM _{l1}	8.3	6.21	13.9	26.7	12	18.6
C4.5	10.52	6.84	14.6	38.3	23.1	23.95
Machete	9.6	5.63	12.76	25.5	21.2	22.3
Scythe	7.6	5.06	12.15	25	16.3	18.1
DANN	8.55	4.92	11.4	30.1	9.7	19.23
Adamenn	7.8	4.78	11.6	26.2	7.7	17.2
WkNN	7.92	4.88	11.3	26.67	8.14	17.8

Table 2.2 Comparisons on classification error on news group (%)

Dataset	K-means	Girolami	SVC	NJW	NSVC	Search-NSVC
(1)	12.0	15.8	12.9	17.7	13.1	14.4
(2)	13.79	15.8	14.2	15.92	11.07	8.03
(3)	7.5	6.4	4.96	8.1	4.92	5.3
(4)	5.95	5.1	5.3	7.3	6.5	5.2
(5)	5.8	3.2	3.98	6.4	5.83	7.74
(6)	4.9	7.8	5.9	6.55	4.72	7.68

information. K-means's behavior is moderate since it is heavily affected by data distribution, and depends on the hard partition classification idea.

The paper considers Facebook-Dataset, which is taken from Max Plank institute for software systems (<http://socialnetworks.mpi-sws.org/data-wosn2009.html>). This dataset includes two classes and the size of dataset equal to 57 kB. Each line of dataset contains of two unknown user identifiers, where the second user posts on the first user's Facebook wall. The other attributes are the number of frequent message, frequent post, frequent business, and frequent application. The number of samples equal to 2,700, which is divided to 2,430 (Training data = $(2,700/10) \times 9$) instances as a training data and 270 (Test data = $2,700 - 2,430$) as a test data (Table 2.1). This dataset includes six attributes as listed in Table 2.2. All three-classification methods are implemented in Weka 3.7.4 software. The system requirements are Weka 3.7.4 software on Mac OS X version 10.6.8 with processor 2.4 GHz Intel Core 2 Duo, memory 4 GB, and 1,067 MHz DDR3 for comparing the percentage of accuracy of classifications. We take some data to form the experimental subsets: (1)–(6).

2.6 Conclusion

This paper presents a simple approach to estimate SVM output confidence. Using the confidence value as weights, WSVM schema works well in practical classification problems. AkNN deals with the difficult cases rejected by WSVM. It employs an informative metric to develop neighborhood. The hyper parameters are auto learned which facilitates computation. Experiments on real datasets evidence fine performance and efficiency of WSVM.

References

1. Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines, vol 10. Cambridge University Press, London, pp 381–388
2. Richard MD, Lippmann RP (1991) Neural network classifiers estimate bayesian a posteriori probabilities. *Neural Comput* 1(3):461–483
3. Craven M, DiPasquo D, Freitag D (1998) Learning to extract symbolic knowledge from the World Wide Web. In: *Proceedings of 15th national conference on artificial intelligence*, 8:880–886
4. Milgram J, Mohamed CSR (2005) Estimating accurate multi-class probabilities with support vector machines. In: *Proceedings of IEEE international joint conference on neural networks*, 3:1906–1911
5. Kwork TJ (1999) Moderate the outputs of support vector machine classifiers. *IEEE Trans Neural Networks* 10:1018–1032
6. Friedman JH (1994) Flexible metric nearest neighbor classification. In: *Technical report, department of statistics*, vol 6. Stanford University, pp 95–104



<http://www.springer.com/978-1-4471-4804-3>

Informatics and Management Science VI

Du, W. (Ed.)

2013, XXIV, 817 p., Hardcover

ISBN: 978-1-4471-4804-3