Construct Offline and Online Membership Functions Based on SVM

Xiangsheng Rong[1] Ping Ling[2] Ming Xu[1]

[1]Xuzhou Air Force College of P. L. A, Xuzhou 221000, P. R. China
[2]School of Computer Science, Xuzhou Normal University, Xuzhou 221116, P. R. China

Abstract

The classification algorithm presented in this paper consists of Offline and Online Membership Functions, named as OOMF. They cooperated with each other to provide qualified class label of confidence. The offline membership function is derived from decision functions yielded by a weighted SVMs approach (WSVM). The online membership function works in the scenario where offline membership function is of low discrimination. And it is designed by a new kNN (NkNN) that is encoded with a class-wise metric. Some strategies bring computational ease: hyper parameters concerned are tuned context-dependently; training dataset is reduced by a tuning support vector clustering (TSVC); and working set of NkNN is pre-specified. We describe experimental evidence of classification performance improved by our schema over state of the arts on real datasets.

Keywords: Offline and online membership function, SVM, Weighted schema, Parameter tuning

1. Introduction

Common fuzzy classifiers predict data label according to membership functions [1]. Its flexibility to assign data belonging to multi classes with different degrees makes this kind of methods popular in many applications. But much priori knowledge is required to define traditional membership functions (MFs). This paper proposes a framework by combining offline MFs and online MFs (OOMF). These two types of MF are constructed by two hard classifiers, with aim to take advantage of their well-founded analysis procedure to improve classification accuracy. The two hard models are WSVM and NkNN, stemming from 1-vs-r SVMs [2]-[3] and kNN [4]. WSVM modifies SVMs into a weighted version, and weights of basic classifiers are integrated with decision function values to define offline MFs. NkNN explores query's neighborhood under the guidance of a class-wise metric. This metric is derived from SVM decision interfaces; for they hold most dicriminant direction along which data are well separated. For a query, it is tested by offline MFs firstly. If the decision is not confident sufficiently, the query will be addressed by online MFs. Online MFs are developed by integrating neighborhood size and distance from query and class.

OOMF uses some strategies to save computation. Firstly, dataset is reduced by a Support Vector Clustering (SVC) [5]. But in this paper Kernel scale of SVC is tuned adaptively, so named as TSVC. Secondly, hyper parameters concerned with support-vector procedure are learned from data context. Thirdly, a heuristic is presented to specify the size of the neighborhood where NkNN works. Experiments on real datasets demonstrate the better performance of OOMF over the state-of-the-art fuzzy classification methods and other popular classification approaches.

2. Related Knowledge

Firstly, we review SVM. For l samples: (x_1, y_1), $(x_2, y_2)\ldots(x_l, y_l)$ sampling from $X\times Y$, where $X = R^n$, $Y = \{1, -1\}$. The optimal classification interface is determined by:

$$g(x) = \sum_i \alpha_i y_i K(x_i, x) - b \qquad (1)$$

The orientation vector α and offset vector b are obtained by optimizing:

$$\max_{\alpha} \ \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j K(x_i, x_j) \qquad (2)$$

$$\text{s.t. } 0 \le \alpha_i \le C_{svm}, \ \sum_i \alpha_i y_i = 0 .$$

Points with $0 < \alpha_i < C_{svm}$ are nbSV. bSV is points with $\alpha_i = C_{svm}$.

k nearest neighbor (kNN) finds query's k nearest neighbors and predicts it as the most frequent one occurring of neighbors. This paper uses it to deal with the case that memberships of all classes are below some threshold.

Then it proceeds to SVC. It finds the smallest hyper sphere containing all data. The produced nbSVs form cluster contours. It corresponds to below optimization:

$$\max_{\gamma} \quad \Sigma\gamma_i K(x_i, x_i) - \Sigma\Sigma\gamma_i\gamma_j K(x_i, x_j) \qquad (3)$$
$$\text{s.t.} \quad \sum_i \gamma_i = 1, \ 0 \le \gamma_i \le C_{svc}$$

3. OOMF Schema

For the M-classification problem, OOMF does:

1) Reduce training dataset using TSVC;
2) Create SVMs and confidence weights $\{\beta_{IA}\}$;
3) Define offline MFs: h_j (j = 1…M) based on SVMs decision functions and $\{\beta_{IA}\}$;
4) $h_{\max}$ = max { h_j };
5) $h_{\sec}$ = second-max { h_j };
6) If $(h_{\max} - h_{\sec}) < \varepsilon$
7) Formulate class-wise metrics;
8) Create online MFs h_j with NkNN;
9) Else label(q) = $h_{\max}$ (q);

The decision whether the result of offline MFs is confident or not, is controlled by threshold ε, which is he difference between its top value of and the second value. In this paper, ε is set as $0.3 \cdot h_{\max}$.

3.1. TSVC

TSVC is conducted on each class respectively to extract data representatives. C_{svc} is set as 1. For point x TSVC sets its scale factor $\sigma_x = \|x\text{-}x_r\|$. Affinity between x and y is scaled by their scale factors' product, that is:

$$k(x, y) = \exp(-\frac{\|x-y\|^2}{\sigma_x \cdot \sigma_y}) = \exp(-\frac{\|x-y\|^2}{\|x-x_r\| \cdot \|y-y_r\|}) \qquad (4)$$

x_r is the r^{th} nearest neighbor of x. For the given r, if $\|x\text{-}x_r\| < \|y\text{-}y_r\|$, it means the density of x's neighborhood is denser than that of y. Here, r is set as the max gap in the list of distances from x to other points: $r = \max_j \{d(x, x_j) - d(x, x_{j-1})\}$. Rows of Euclidean distance matrix $d(x, x_j)$ are sorted in an ascending order.

This tuning produces more nbSVs than traditional SVC. These nbSVs are located on both boundaries and important positions where sharp variance of density happens. So an informative sketch of dataset is described by nbSVs, which act as data representatives. Fig. 1 and 2 show results of the tuning SVC and fixed-scaled SVC.

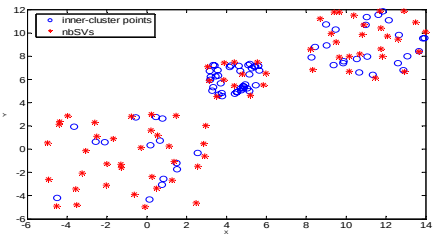

Fig. 1: Tuning SVC.

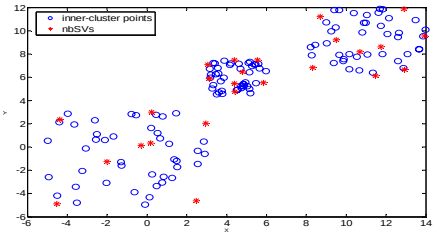

Fig.2: SVC with 1/σ2 = 0.537.

3.2. Offline MFs and WSVM

Offline MFs are defined during WSVM training process. Different from traditional SVMs, where basic classifiers are weighted equally, WSVM seeks coefficients β_{IA} for basic SVM (class I-vs-r) (I = 1…M) to show its decision capacity on class A. Weights of all SVMs form a matrix $\beta = (\beta_{IA})_{M \times M}$. Set $\beta_{II} = 1$, which is natural that SVM (I-vs-r) is absolutely confident to declare query's membership to class I. For point x, it corresponds to a series of function values with respect to basic SVM (I-vs-r), and these values from a row vector: $F_x = (f_1(x), f_2(x) \ldots f_M(x))$, where f_I is the decision function of basic SVM (I-vs-r). The offline MF with respect to class A, $h_A(x)$, is developed as:

$$h_A(x) = F_x \cdot \beta_{\bullet A} \qquad (5)$$

With $$\beta_{\bullet A} = \Sigma_{I=1}^{M} \beta_{IA} \ \beta_{IA} \ge 0 \qquad (6)$$

Weight β_{IA} is designed based on the distance from f_I to class A:

$$\beta_{IA} = \begin{cases} 1 & I = A \\ -\dfrac{\exp[dis(Center\_A, f_I)]}{\Sigma_{J=1, J \ne I}^{M} \exp[dis(Center\_J, f_I)]} & I \ne A \end{cases} \qquad (7)$$

Exponential mechanism is used to keep β_{IA} stable. Minus in $I \neq A$ case is introduced to match the negative value of $f_I(x)$ when x belongs to rest classes except I. $dis(Center\_A, f_I)$ computes the distance between the center of class A and f_I, as shown in (8), where $Center\_A$ is the average of A's data representatives. The first term is the margin of f_I, where $\|w\|_I$ is the weight vector of f_I. The second term is the distance from class center to the nearest nbSV of f_I.

$$dis(Center\_A, f_I) \approx \frac{1}{\|w_I\|^2} + \min_{s \in \{nbSVs\}} \| Center\_A - s \| \qquad (8)$$

3.3. Basic SVM Construction

Tune SVM Kernel scale. Firstly for class I, its Kernel scale factor is designed as:

$$\tau_I = ave\{\| x - x_r \|\} \quad \text{with} \quad x \in I \qquad (9)$$

Gaussian Kernel of SVM I-vs-r sets its scale as:

$$\sigma_I^2 = \tau_I \cdot \tau_{rest(I)} \text{ with } \tau_{rest(I)} = ave\{\tau_J \mid J \ne I\} \qquad (10)$$

Tune penalty parameter individually. This paper equips the individual C_{svm} for each point to express its individual demand for slack variable. To those outliers or bSVs, they hope a big C_{svm} to emphasis the slack, but to inner-class-points, they need a small one to highlight maximum margin. Therefore we set $C_{svm(x)}$ for x in (11).

$$C_{svm(x)} = \frac{||x - x_r||}{r} \tag{11}$$

Check tuning effect. We perform tuned SVM and standard SVM on real datasets [6]. Standard SVM sets its hyper parameters by 5-fold cross validation. From Table 1, tuned SVM gives competitive results with the optimal results of standard SVM, while consuming less computation time. This indicates the quality of tuning strategies.

Data	tuned SVM		standard SVM	
	Error(%)	Time(s)	Error(%)	Time(s)
Iris (class 1-vs 2, 3)	0	0.672	0	1.108
Iris (class 2-vs 1, 3)	4.2	0.702	4.1	1.131
Iris (class 3-vs 1, 2)	3.27	0.691	3.26	1.107
Breast Cancer	2.52	3.87	2.41	7.05

Table 1: Classification accuracies and time cost comparison on the average of 20 runs. (%) (20% data are randomly sampled for training).

4. Online MFs and NkNN

4.1. Define Online MFs from NkNN

NkNN considers sub neighborhood in each class A: $sNEI_A$. $sNEI_A$ is developed under the guidance of the metric customized to A. This metric also helps to compute the distance between query and A. Sub neighborhood size is taken as class frequency and the distance as the weight. Denote the class frequency $t_A = |\, sNEI_A \,|$. Let $Cd_A(x, A)$ be the distance between x and class A, then online MF is defined as:

$$h_A(x) = \left(1 - \frac{Cd_A(x,\ A)}{\Sigma_{I=1}^{M}\ Cd_I(x,\ I)}\right) \cdot t_A \tag{12}$$

4.2. Class-Wise Metric

Class-wise metric is expected to reveal more of class's intrinsic data features, and consequently produce small inner-class distance values and big inter-class ones. Another reason to define a new metric is to overcome the curse-of-dimensionality that all kNN-based methods have to deal with. The new metric is derived from SVM decision interface, the byproduct of WSVM, which facilitates computation greatly. For decision function of SVM (A-vs-r), f_A, viewed under theoretical light, it is optimal in the sense of structural risk minimization. Viewed from geometry light, to point x on level curve $f_A(x) = 0$, the gradient vector $f_A'(x)$ indicates the perpendicular orientation along which data can be well separated over x's neighborhood. That is, $f_A'(x)$ tells the local relevance of input features in the sense of identifying class A.

We probe a representative point P_A from class A to generate the discriminant direction with respect to A. The point closest to curve f_A is selected as P_A. Clearly, P_A comes from support vector set, so it can be found by following optimization:

$$\min_x \ f_A(x), \qquad \text{with} \quad x \in A. \tag{13}$$

Denote $f_A'(P_A) = g_A = (g_{A1}, g_{A2}, \ldots, g_{An})$. Then magnitude of each component reveals the importance of the corresponding dimension when identifying class A. Based on this idea, class-wise metric special for class A is defined as:

$$Cd_A(x, y) = \sqrt{(x-y)^T \mu^A (x-y)} \tag{14}$$

$$\mu_i^A = \frac{\exp(|g_{Ai}|)}{\Sigma_{j=1}^{n} \exp(|g_{Aj}|)} \tag{15}$$

Introduce the center of $sNEI_A$ of class A, $x^{(A)}$, to yield the distance from x to A:

$$Cd_A(x, A) = \sqrt{(x-x^{(A)})^T \mu^A (x-x^{(A)})} \tag{16}$$

4.3. $sNEI_A$ Specification

The investigation of $sNEI_A$ is pricy. This paper uses the max gap of distance information from query to A-class members to set that size. That is:

$$t_A = \max_j \left\{ \frac{Cd_A(Q,x_{j+1}) - Cd_A(Q,x_j)}{Cd_A(Q,x_j) - Cd_A(Q,x_{j-1})} \ \middle|\ x_j \in A \right\} \tag{17}$$

Here distance list $Cd_A(Q, x_j)$ is sorted in the ascending order. Set $Cd_A(Q, x_0) = 0$, and $Cd_A(Q, x_1) = 0$, then t_A tells the number of $sNEI_A$ members including Q itself.

5. Experimental Results

Data	SVM_{1r}	SVM_{11}	FCM	FSVM	OOMF
Thyroid	4.37	4.42	4.92	5.14	4.26
Heart	5.76	6.14	5.27	7.11	5.31
Diabetes	11.12	12	10.9	11.2	8.14
Wine	28.4	26.2	27.3	26.5	30.67
Waveform	3	2.6	3.6	3.3	3
Liver	8.12	8.3	8.2	7.6	7.92

Table 2: Comparison on classification error (%) (30% data are sampled randomly for training.)

First six datasets are taken from UCI Machine Learning Repository [6]. In Table 2, OOMF is compared with two SVM-based classifiers: SVMs of 1-vs-r version (SVM_{1r}) [7] and SVMs of 1-vs-1 version (SVM_{11}) [8]; and two fuzzy classifiers: FCM

[9], and FSVM [10]. These classifiers set hyper parameters by cross validation.

Compared with two SVM-based classifiers, OOMF's improvement is obvious due to its soft decision fashion. Two SVM-based approaches are competitive. In three of six datasets OOMF achieves best result and this behavior is better over two fuzzy classifiers, which comes from the employment of local MFs. OOMF behaves not so well in Wine dataset, because in this set 178 data cover 13 dimensions and the neighborhood information is too weak to facilitate online MF's job. For two fuzzy methods, FSVM exhibits better behaviors on average. It relies on the nonlinear decision interface produced by SVM, while FCM depends on regular partition based on Euclidean metric, so the lack of adaptation to datasets leads to high error ratios.

Then OOMF is performed on News group [11]. This dataset is a compilation of about 20,000 articles (email messages) evenly divided among the 20 categories like religion, politics and sports. We label each newsgroup as follows:

NG1: alt.atheism; NG2: comp.graphics; NG3: comp.os.ms.windows.misc; NG4: comp.sys.ibm.pc.hardware; NG5: comp.sys.mac.hardware; NG6: comp.windows.x; NG7: misc.forsale; NG8: rec.autos; NG9: rec.motorcycles; NG10: rec.sport.baseball; NG11: rec.sport.hockey; NG12: sci.crypt; NG13: sci.electronics; NG14: sci.med; NG15: sci.space; NG16: soc.religion.christian; NG17: talk.politics.guns; NG18: talk.politics.mideast; NG19: talk.politics.misc; NG20: talk.religion.misc.

We apply *tf.idf* weighting schema to express documents. We delete the stop words and words that appear too few times, and then normalize each document vector to have unit Euclidean length. Some other approaches are considered on the average classification error rates of 10 runs: Simple *k*NN; C4.5 decision tree [12]; Machete [4], a recursive partitioning procedure, where the dimension used for splitting at each step is the one that maximizes the estimated local relevance; Scythe [4], a generalization and modification of Machete method; DANN, an adaptive nearest neighbor method [13]; and Adamenn, another adaptive nearest neighbor approach proposed in [14]. For the clearness of table, these methods are denoted as: a) *k*NN, b) SVM_{1r}, c) SVM_{11}, d) C4.5, e) Machete, f) Scythe, g) DANN, h) Adamenn, i) OOMF. We sample from some classes to form experiment subsets, and these subsets are listed below, where numbers in bracket are sampling size. Experiment results are recorded in Table 3.

1) {NG2, NG3, NG4} (300)
2) {NG2 (150), NG3 (50), NG4 (200)}
3) {NG6, NG7, NG8} (150)
4) {NG7 (200), NG8 (150), NG9 (350)}
5) {NG1, NG2, NG7, NG8} (200)
6) {NG1 (50), NG2 (100), NG7 (150), NG8 (50)}
7) {NG7 (100), NG8 (50), NG12 (200), NG16 (50), NG17 (100)}

	a)	b)	c)	d)	e)	f)	g)	h)	i)
1)	32.0	30.8	31.9	37.7	33.1	34.4	35.8	30.2	30.2
2)	33.7	30.8	31.2	34.9	31.0	29.0	31.4	31.2	31.2
3)	17.5	16.4	15.9	17.1	15.9	15.3	16.7	15.7	16.1
4)	15.9	15.1	16.3	18.3	16.5	15.2	14.8	14.5	14.3
5)	15.8	13.2	12.9	14.4	13.8	13.7	13.6	12.5	12.6
6)	14.9	13.8	13.9	13.6	13.7	12.6	13.0	13.8	13.6
7)	15.6	12.3	12.3	14.7	12.9	12.0	11.6	12.5	12.2

Table 3: Classification error comparison on News Group (%)

From Table 3, it finds that in the subsets where class boundaries are not distinct, like {NG2, NG3, NG4}, {NG8, NG9, NG10}, OOMF shows a unique good job among its peers. This is attributed to the contribution of local MFs to identify data's label context-dependently. In the subsets where class boundaries are distinct, OOMF yields steady and moderate results and usually its result follows secondly the optimal result. Here, SVM_{1r} does a better job than SVM_{11}. C4.5 and Machete work poorly in some sets due to their greedy idea. Scythe modifies the greedy nature used by Machete and thereby achieves higher accuracy. DANN work well, but the metric it employs approximates the weighted *Chi-squared* distance, which will causes its failure in datasets of non-Gaussian distribution. Adamenn works well in most cases, but it requires huge cost to tune six parameters. If cost is considered, OOMF is a fine choice.

6. Conclusions

A fuzzy classification algorithm OOMF is described in this paper. Its offline MFs are defined by WSVM. WSVM modifies SVMs schema by equipping basic classifier with decision weights, which are integrated into decision function to form MFs. If the offline model presents poor confidence, the online MFs are created by NkNN. NkNN also use a weighted strategy to do label assignment. The neighborhood where NkNN works is formulated under the class-wise metric that is derived from SVM decision function. Training dataset size is reduced and hyper parameters are learned data-dependently, which bring computational benefit. Experiments on real datasets evidence fine performance and efficiency of OOMF.

References

[1] T. Inoue, S. Abe, Fuzzy support vector machine for pattern classification, *Proc. of International Joint Conference on Neural Networks*, pp. 1449-1455, 2001.

[2] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines, Cambridge University Press, London, 2000.
[3] L. P. Wang, Support Vector Machines: Theory and Application, Springer, Berlin Heidelberg New York, 2005.
[4] J. H. Friedman, Flexible Metric Nearest Neighbor Classification, *Tech. Report, Dept. of Statistics*, Stanford University, 1994.
[5] A. Ben-Hur, D. Horn, H. T. Siegelmann, „Support Vector Clustering, *Journal of Machine Learning Research* pp. 125-137, 2001.
[6] http://www.uncc.edu/knowledgediscovery.
[7] V. Vapnik, Statistical Learning Theory, Wiley New York, 1998.
[8] T. J. Hastie, R. J. Tibshirani, Classification by Pairwise Coupling. *Advances in Neural Information Processing Systems*, MIT Press, Cambridge, 10:507-513, 1998.
[9] J. C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, New York, Plenum Press, 1981.
[10] H. Han-Pang, L. Yi-Hung, Fuzzy Support Vector Machines for Pattern Recognition and Data Mining, *Fuzzy Systems* , 4(3):826-835, 2002.
[11] http://www.cs.cmu.edu/afs/cs/project/theo-11/www/naive-bayes.html.
[12] J. R. Quinlan, C4.5: Programs for Machine Learning, Morgan-Kaufmann Publishers, 1993.
[13] T. Hastie, R. Tibshirani, Discriminant Adaptive Nearest Neighbor Classification, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 18(6):607-615, 1996.
[14] C. Domeniconi, J. Peng, D. Gunopulos, An Adaptive Metric Machine for Pattern Classification, *Advances in Neural Information Processing Systems*, 13, 2000.