Multi-class Classification in Big Data

Anton Malenichev[1], Olga Krasotkina[1], Vadim Mottl[2], and Oleg Seredin[1]

[1] Tula State University, Tula, Russia
malenichev@mail.ru, o.v.krasotkina@yandex.ru, oseredin@yandex.ru
[2] Dorodnicyn Computing Centre, RAS, Moscow, Russia
v.v.mottl@yandex.ru

Abstract. The paper suggests the on-line multi-class classifier with a sublinear computational complexity relative to the number of training objects. The proposed approach is based on the combining of two-class probabilistic classifiers. Pairwise coupling is a popular multi-class classification method that combines all comparisons for each pair of classes. Unfortunately pairwise coupling suffers in many cases from incompatibility in that some regions of its input space the sum of probabilities are not equal to one. In this paper we propose the optimal approximation for probabilities in each point of object space. This paper proposes a new probabilistic interpretation of the Support Vector Machine for obtaining class probabilities. We show how the SVM can be viewed as a maximum likelihood estimate of a class of probabilistic models. As a computational method for big data we use the stochastic gradient descent approach minimizing directly the primal SVM objective. Unfortunately the hinge loss of the true SVM classifier did not allow to use SGD procedure for determining the classifier bias. In this paper we propose the piece-wise quadratic loss that helps to overcome this obstacle and gives an instrument to obtain the bias from SGD procedure.

Keywords: pairwise coupling, stochastic gradient descent, support vector machine, multiclass learning, large datasets.

1 Introduction

The multi-class classification problem refers to assigning each of the observations into one of k classes. Most of the real world classification applications, such as image search and text recognition, involve many classes. Thus, the user needs to select from amongst a large class of labels in addition to handling a huge data set.

In the general sense the multi-label classification methods can be categorized into two main categories: "Single Machine approaches" that try to solve a single optimization problem that trains many binary classifiers simultaneously and approaches based on combining independent binary classifiers. Weston and Watkins [1] propose a formulation of Support Vector Machine approach that enables a multi class problem to be solved in a single optimization criterion. Lee, Lin and Wahba [2] propose the multicategory support vector machine (MSVM),

which extends the binary SVM to the multicategory case. The proposed method provides a unifying framework when there are either equal or unequal misclassification costs. Both Weston and Lee formulations have many dummy variables, have no decomposition method and could not be used in the case of big training samples. Crammer and Singer [3] describe the algorithmic implementation of multiclass kernel-based vector machines with efficient iterative decomposition scheme. But proposed algorithm use a lot of memory to cache kernel products. The paper states that the fastest version has "two more technical improvements which are not discussed here but will be documented in the code that we will shortly make available". But the code was never made available.

The dominating approach for solving multiclass problems has been based on reducing a single multiclass problems into multiple binary problems. Constructing of all-versus-all (AVA) or one-versus-all (OVA) classifiers is a popular approach in this strategy [4,5,6]. As can be seen from the literature, AVA seems faster and more memory efficient in the case of big sample. It requires $O(m^2)$ classifiers instead of $O(m)$ (m - number of classes), but each classifier is (on average) much smaller. If the time to build a classifier is superlinear in the number of data points, AVA is a better choice.

A common way to combine pairwise comparisons is by voting [8,9]. It constructs a rule for discriminating between every pair of classes and then selecting the class with the most winning two-class decisions. The voting procedure only predicts a class label. In many cases, however, probability estimates are desired. Hastie and Tibshirani [10] have proposed probability estimates by combining the pairwise class probabilities. In this paper we propose a method for combining the class probabilities that is more stable than voting and the method by Hastie and Tibshirani.

This paper proposes a new probabilistic interpretation of the Support Vector Machine for obtaining class probabilities. We show how the SVM can be viewed as a maximum likelihood estimate of a class of probabilistic models. As a computational method for big data we use the stochastic gradient descent approach minimizes directly the primal SVM objective. Unfortunately the hinge loss of the true SVM classifier did not allow to use SGD procedure for determining the classifier bias. The bias term often plays a crucial role when the distribution of the labels is uneven as is typically the case in text processing applications. Shalev-Schwarz et al. [17] proposed several approaches for learning the bias term. This approach simply amounts to adding one more feature to each instance and incorporating the bias into direction vector. The disadvantage of this approach is that we solve a relatively different optimization problem, not SVM. Second approach consist in optimizing the non-convex SVM loss as is. But in this case the algorithm has slower convergence rate. In this paper we propose the piecewise quadratic loss that helps to overcome this obstacle and gives an instrument to obtain the bias from SGD procedure.

The paper is organized as follows. Section 2 states the optimal approximation for pair-wise probabilities defining the pairwise coupling approach. Section 3

reviews the binary SVM. Section 4 presents a numerical study for illustration. Then, Section 5 presents concluding remarks and discussion of future directions.

2 The problem of multi-class classification on large data sets

The paper considers the classical formulation of the learning pattern recognition problem. We suppose that there are many real-world objects Ω. We also assume that each object $\omega \in \Omega$ may be characterized by the label $y \in \{0, 1, .., m-1\}$, $m > 2$, and by vector $\mathbf{x} \in R^n$. In other words, each object is most fully represented by triple $(\omega_i, y_i, \mathbf{x}_i)$, $i = 1, .., N$, where N is the count of objects in current set. The number y_i in this case is called the class of the object ω_i, vector $\mathbf{x}_i$ is a feature vector. The number n, is the dimension of the feature space, indicates the length of vectors $\mathbf{x}_i$. Usually we often know only the feature vector $\mathbf{x}_i$ for each object ω_i, and doesn't know its class y_i. The recognition problem is to build a function, which requires the feature vector $\mathbf{x}_i$ and returns the class of an object $\hat{y}_i$, and it has to make mistakes as little as possible.

This problem is unsolvable without some additional information about objects. Suppose that we have some set of objects $\omega_i^{train}, i = 1, .., N$, for which a feature vector $\mathbf{x}_i$ is known as well as their class labels y_i. The whole set of such objects called the training set that consists of N objects. Say at once that the number N in the case of the large data sets analysis is large enough. For clarity we determine $N \geq 10000$.

2.1 Our pairwise coupling approach

It is easy to calculate that if we have m different classes the number of different pairs of classes equals to

$$C_m^2 = \frac{m(m-1)}{2}. \tag{1}$$

Suppose that we want to determine the pairwise confidence function $P^{kl}(\mathbf{z})$. Let us select objects of the classes k and l from the initial training set. Objects of other classes are not considered when constructing pairwise confidence function. Note that it's necessary to make calculation only for $k < l$. If $k = l$ the problem does not make sense, and if $k > l$ we can make a simple and obvious transition $P^{lk} = 1 - P^{kl}$.

After receiving whole set of pairwise confidence functions $P^{kl}(\mathbf{z})$ it is requires to construct the general classification rule $\pi(\mathbf{z})$.

Suppose that at the point $\mathbf{z}$ the required distribution $\pi(\mathbf{z})$ exists and it is agreed with all pairwise probabilities $P^{kl}(\mathbf{z})$. Actually, strictly speaking, this is not always true, since pairwise probabilities $P^{kl}(\mathbf{z})$ are obtained independently. Besides their combination may be inconsistent. However, experience shows that if the inconsistency observed, it presents in very small areas of the feature space only. Moreover, we show how to choose the approximation in case of inconsistency.

Suppose that each pairwise probability is equal to

$$P^{kl}(\mathbf{z}) = \frac{\pi^k(\mathbf{z})}{\pi^k(\mathbf{z}) + \pi^l(\mathbf{z})}, \tag{2}$$

$$P^{lk}(\mathbf{z}) = 1 - P^{kl}(\mathbf{z}) = \frac{\pi^l(\mathbf{z})}{\pi^k(\mathbf{z}) + \pi^l(\mathbf{z})}. \tag{3}$$

As a simplification we assume also that

$$P^{kk}(\mathbf{z}) = 0.5. \tag{4}$$

Let express $\pi^l(\mathbf{z})$ from (2)

$$\pi^l(\mathbf{z}) = \frac{1 - P^{kl}(\mathbf{z})}{P^{kl}(\mathbf{z})}\pi^k(\mathbf{z}),\ l \neq k. \tag{5}$$

Since whole set of $\pi(\mathbf{z})$ makes a complete group of events we can express $\pi^k(\mathbf{z})$

$$\sum_{k=1}^{m} \pi^k(\mathbf{z}) = 1 \tag{6}$$

$$\pi^k(\mathbf{z}) = 1 - \sum_{\substack{l=1 \\ l \neq k}}^{m} \pi^l(\mathbf{z}) = 1 - \left(\sum_{\substack{l=1 \\ l \neq k}}^{m} \frac{1 - P^{kl}(\mathbf{z})}{P^{kl}(\mathbf{z})} \right) \pi^k(\mathbf{z}). \tag{7}$$

$$\pi^k(\mathbf{z}) = \left(1 + \sum_{\substack{l=1 \\ l \neq k}}^{m} \frac{1 - P^{kl}(\mathbf{z})}{P^{kl}(\mathbf{z})} \right)^{-1}. \tag{8}$$

The formula (8) can be simplified. If we assume that the probability of assigning an object to its own class is 0.5, it becomes

$$\pi^k(\mathbf{z}) = \left(\sum_{l=1}^{m} \frac{1 - P^{kl}(\mathbf{z})}{P^{kl}(\mathbf{z})} \right)^{-1}. \tag{9}$$

If we estimate the dichotomous probability only for $k < l$, then we can rewrite the formula (9) as follows

$$\pi^k(\mathbf{z}) = \left(\sum_{l=1}^{k-1} \frac{1 - P^{lk}(\mathbf{z})}{P^{lk}(\mathbf{z})} + 1 + \sum_{l=k+1}^{m} \frac{1 - P^{kl}(\mathbf{z})}{P^{kl}(\mathbf{z})} \right)^{-1}. \tag{10}$$

We have to note that if at least one of the probabilities P^{kl} in the original formula (8) is zero, then the corresponding denominator becomes zero too.

Consider an infinitesimal value $P^{kq} \to +0$, $q \neq k$. Then the formula (8) can be rewritten using the limit

$$\lim_{P^{kq}\to+0} \pi^k(\mathbf{z}) = \lim_{P^{kq}\to+0} \left(\frac{1}{P^{kq}(\mathbf{z})} \right)^{-1} = \lim_{P^{kq}\to+0} \left(P^{kq}(\mathbf{z}) \right) = 0. \qquad (11)$$

In other words, if at least one of the dichotomous probabilities $P^{kl}(\mathbf{z}) = 0$, $k \neq l$, the corresponding probability $\pi^k(\mathbf{z}) = 0$.

The inconsistency is reflected in the fact that equality (6) is not satisfied, i.e. the sum of all the probabilities is not equal to one. The easiest way to find the approximation is to use the normalisation as follows:

$$\pi^k(\mathbf{z}) = \left(\sum_{l=1}^{k-1} \frac{1 - P^{lk}(\mathbf{z})}{P^{lk}(\mathbf{z})} + 1 + \sum_{l=k+1}^{m} \frac{1 - P^{kl}(\mathbf{z})}{P^{kl}(\mathbf{z})} \right)^{-1} \cdot \left(\sum_{k=1}^{m} \pi^k(\mathbf{z}) \right)^{-1}. \qquad (12)$$

3 Two-class classification: Optimizing primal Support Vector Machine objective

We are using a linear decision function for classification. Let us assume that there is a hyperplane, which correctly classifies almost all objects from the training sample $(X, Y) = \{(x_j, y_j),\ j = 1, ..., N\}$, $d(x_j \,|\, a, b) = (a^T x_j + b)$ for all $j = 1, ..., N$

The loss function in general looks as follows:

$$q(x, y, a, b) = \{\max[0,\ \ 1 - yd(x, a, b)]\}^{\alpha}. \qquad (13)$$

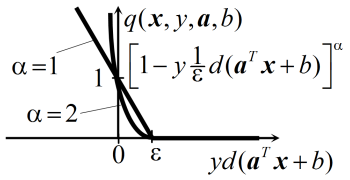

Fig. 1. The view of loss functions for different values of α

In the original formulation of the support vector machine the degree α shall to be equal to one. However, this leads to fracture of the loss function and, consequently, nondifferentiability at the break point. In this paper we use the gradient method, which involves the procedure of differentiation of the original SVM criterion and contains the sum of loss function values for all objects lying in the area between the hyperplane and the gap. It's so-called support objects, and vector features, describing these objects, are called the support vectors. We need the loss function which is differentiable at all points, so we take $\alpha = 2$:

$$q(x, y, a, b) = \{\max[0,\ \ 1 - yd(x, a, b)]\}^2. \tag{14}$$

We'll choose a hyperplane for which the gap between it and the nearest vector of training set in the sense of the Euclidean metric in R^n is maximum

$$y_j d(x_j \,|\, a, b) \ = \ y_j(a^T x_j \ + b) \ \geq \ \varepsilon, \ \varepsilon \to \max, \ a^T a \ = \ 1. \tag{15}$$

This formulation of the problem leads to the following criteria:

$$J(\mathbf{a}, b) = \mathbf{a}^T\mathbf{a} + C \sum_{j:y_j(\mathbf{a}^T\mathbf{x}_j+b)\leq 1} \left[1 - y_j(\mathbf{a}^T\mathbf{x}_j + b)\right]^2 \to \min(\mathbf{a}, b). \tag{16}$$

Usually an SVM criterion is optimized in a dual form. It gives the exact solution, but the high computational complexity and the need of loading the whole training objects into the memory at the same time prohibit the using of this method on a large training sets. We need the method for online learning, which would produce the adjustment of decision rule over the time on the basis of single or few random training objects for each iteration. In this paper we propose to solve the primal SVM objective using an iterative approximation method of stochastic gradient descent. It allows the on-line training without loading the entire training set to the memory.

There are several implementations of stochastic gradient descent methods for solving the primal SVM objective [11,13,14,17]. However, they only allow us to estimate the normal vector of hyperplane, while ignoring the bias value. In order to estimate the bias methods based on ROC-analysis are usually used, which significantly affects the final computational complexity and eliminates the advantage in speed. This paper proposes the method for optimizing the original SVM criterion with a quadratic loss function using the stochastic gradient descent method. This simple method combines high performance, capacity for additional training and simultaneous assessment of the normal vector and bias of the hyperplane.

Let us go to the expanded feature space via introducing new designations:

$$\begin{cases} \mathbf{c} = \begin{pmatrix} \mathbf{a} \\ b \end{pmatrix} \in R^{n+1} \\ \mathbf{A} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{pmatrix} [(n+1) \times (n+1)] \ . \\ \mathbf{z}_j = \begin{pmatrix} \mathbf{x}_j \\ 1 \end{pmatrix} \in R^{n+1} \end{cases} \tag{17}$$

According to equations (17) the training criterion can be rewritten as follows:

$$J(\mathbf{c} \in {}^{n+1}) = \mathbf{c}^T\mathbf{A}\mathbf{c} + C \sum_{j:y_j\mathbf{z}_j^T\mathbf{c}\leq 1} \left(1 - y_j\mathbf{z}_j^T\mathbf{c}\right)^2. \tag{18}$$

Let us denote by $\mathbf{c}^s = \begin{pmatrix} \mathbf{a}^s \\ b^s \end{pmatrix} \in R^{n+1}$ an approximation of the solution on the s-th iteration of the algorithm. The next approximation is calculated by the formula $\mathbf{c}^{s+1} = \mathbf{c}^s - \alpha^s g\left(J(\mathbf{c}^s)\right)$.

The coefficients α^s are selected to satisfy the condition

$$\sum_{s=-\infty}^{\infty} \alpha^s = \infty; \sum_{s=-\infty}^{\infty} \left(\alpha^s\right)^2 < \infty. \tag{19}$$

Sub-gradient for a single object is equal to

$$\begin{gathered} g_{\mathbf{c}}\left(J(\mathbf{c})\right) = g_{\mathbf{c}}\left(\mathbf{c}^T\mathbf{A}\mathbf{c} + C\begin{bmatrix}(1-y\mathbf{z}^T\mathbf{c})^2, y\mathbf{z}^T\mathbf{c} \le 1\\ 0, y\mathbf{z}^T\mathbf{c} > 1\end{bmatrix}\right) = \\ 2\mathbf{A}\mathbf{c} + 2C\begin{bmatrix}(-y\mathbf{z} + \underbrace{yy}_{1}\mathbf{z}\mathbf{z}^T\mathbf{c}), y\mathbf{z}^T\mathbf{c} \le 1\\ 0, y\mathbf{z}^T\mathbf{c} > 1\end{bmatrix} = \\ 2\mathbf{A}\mathbf{c} + 2C\begin{bmatrix}(-y\mathbf{z} + \mathbf{z}\mathbf{z}^T\mathbf{c}), y\mathbf{z}^T\mathbf{c} \le 1\\ 0, y\mathbf{z}^T\mathbf{c} > 1\end{bmatrix} = \\ 2\left(\mathbf{A} + C\begin{bmatrix}\mathbf{z}\mathbf{z}^T, y\mathbf{z}^T\mathbf{c} \le 1\\ 0, y\mathbf{z}^T\mathbf{c} > 1\end{bmatrix}\right)\mathbf{c}^s - 2C\begin{bmatrix}y\mathbf{z}, y\mathbf{z}^T\mathbf{c} \le 1\\ 0, y\mathbf{z}^T\mathbf{c} > 1\end{bmatrix} \end{gathered} . \tag{20}$$

The algorithm stops when the condition $\left|J(\mathbf{c}^{s+1}) - J(\mathbf{c}^s)\right| < \xi$, is satisfied. Here ξ - required accuracy.

In this case the posterior probability of belonging to one of the two classes expresses as (21)

$$\begin{gathered} P^{kl} = \begin{cases} \frac{\exp\left[-C\left(1-\mathbf{z}^T\mathbf{c}\right)^2\right]}{1+\exp\left[-C(1-\mathbf{z}^T\mathbf{c})^2\right]}, & \mathbf{z}^T\mathbf{c} < -1, \\ \frac{\exp\left[-C\left(1-\mathbf{z}^T\mathbf{c}\right)^2\right]}{\exp\left[-C(1+\mathbf{z}^T\mathbf{c})^2\right]+\exp\left[-C(1-\mathbf{z}^T\mathbf{c})^2\right]}, & -1 \le \mathbf{z}^T\mathbf{c} \le 1, \\ \frac{1}{\exp\left[-C(1+\mathbf{z}^T\mathbf{c})^2\right]+1}, & \mathbf{z}^T\mathbf{c} > 1, \end{cases} \\ P^{lk} = 1 - P^{kl}. \end{gathered} \tag{21}$$

4 Experimental research

The experimental research was performed on real datasets from UCI repository in opposition to Hastie and Tibshirani method. There were 3 datasets used. Short dataset descriptions are presented at the table 1 as well as the results of experimental study. The timings are presented for pairwise coupling procedures only. The binary classifier for both cases is SGD SVM with pairwised quadratic loss. Bold values mean better results.

The experimental stand consisted of CPU Intel Core i5-2430M 2.4Ghz, 8 Gb RAM. The experiment was performed at single core.

The experimental research shows that proposed approach has low computational complexity as well as low error rate and it fits well for multiclass big data recognition tasks.

Table 1. Summary table of the experimental research

Dataset name	Classes, m	Features, n	# train objects	# test objects	Hastie & Tibshirani method		Proposed method	
					Err, %	Time, s	Err, %	Time, s
Pendigits	10	16	7494	3498	**4.49**	1.46	4.69	**0.39**
Satimage	6	36	4435	2000	15.30	1.01	**15.20**	**0.18**
Kdd-cup	6	112	47120	20191	29.54	9.44	**14.00**	**1.71**

5 Conclusion

Highly effective method for multi-class classification in big data was proposed. It based on the pairwise probability classifiers coupling in accordance to AVA scheme. It's pretty easy to implement, but it has good recognition abilities.

As the binary classifier was proposed the modified Stochastic Gradient Descent SVM method that have sublinear computational complexity relative to the number of training objects. The main drawback of the original SGD SVM method is inability of evaluating the bias as well as the normal vector of hyperplane. To overcome it we proposed to use the piecewised quadratic loss function.

Experimental research shows that developed method successfully handles with the multi-class classification task in big data with acceptable timing and accuracy.

The main direction for future study is further decreasing of computational complexity by using methods for non-enumerative cross-validation based on the classical Akaike Information Criterion.

Acknowledgments The work supported by grants 0018322 of the Foundation for Assistance to Small Innovative Enterprises (FASIE) and 14-07-00964, 16-37-00399, 14-07-00527, 16-57-52042 of the Russian Foundation for Basic Research.

References

1. *Weston, Jason and Watkins, Chris.* Support vector machines for multi-class pattern recognition, ESANN'1999 proceedings - European Symposium on Artificial Neural Networks Bruges (Belgium), 21-23 April 1999, ISBN 2-600049-9-X, pp. 219-224
2. *Yoonkyung Lee and Yi Lin and Grace Wahba.* Multicategory Support Vector Machines, theory, and application to the classification of microarray data and satellite radiance data, Journal of the American Statistical Association, 2004,99,67–81
3. *Crammer, Koby; and Singer, Yoram.* On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines. Journal of Machine Learning Research, 2001, 2, 265–292.
4. *Duan, K. B., Keerthi, S. S..* Which Is the Best Multiclass SVM Method? An Empirical Study. Multiple Classifier Systems. LNCS 3541., 2005, pp. 278–285.

5. *Hsu, Chih-Wei and Lin, Chih-Jen.* A Comparison of Methods for Multiclass Support Vector Machines. 2002, IEEE Transactions on Neural Networks.
6. *Rifkin, R.* MIT—Multiclass Classification, Available online: http://www.mit.edu/ 9.520/spring09/Classes/multiclass.pdf.
7. *Sergey Dvoenko, Vadim Mottl, Oleg Seredin* Multiclass pattern recognition procedure based on pairwise confidence functions for pairs of classes. Izvestija TulGU, series «Computer science, automation, management», volume 2 part 2, 1999, pp. 28-35. (in Russian)
8. *S. Knerr, L. Personnaz, and G. Dreyfus.* Single-layer learning revisited: a stepwise procedure for building and training a neural network. In J. Fogelman, editor, Neurocomputing: Algorithms, Architectures and Applications. Springer-Verlag, 1990.
9. *J. Friedman.* Another approach to polychotomous classification. Technical report, Department of Statistics, Stanford University, 1996.
10. *T. Hastie and R. Tibshirani.* Classification by pairwise coupling. The Annals of Statistics, 26(1):451–471, 1998.
11. *Bordes, Antoine.* New algorithms for large-scale support vector machines. Diss. Université Pierre et Marie Curie-Paris VI, 2010.
12. *Hosmer Jr D. W., Lemeshow S.* Applied logistic regression. New York: John Wiley & Sons, 2004.
13. *John Duchi and Yoram Singer* Online and Batch Learning using Forward Looking Subgradients, 2008. Manuscript.
14. *Kivinen J., Smola A. J., Williamson R. C.* Online learning with kernels. Advances in neural information processing systems. 2001. pp. 785-792.
15. *Menon, Aditya Krishna* Large-scale support vector machines: algorithms and theory. Research Exam, University of California, San Diego (2009): 1-17.
16. *P. Jain, A. Kapoor* Active Learning for Large Multi-class Problems. Proc. the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009: 762-769.
17. *Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro* Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. ICML 2007: Proceedings of the 24th International Conference on Machine learning, pages 807-814, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-793-3.
18. *Weston J., Watkins C.* Multi-class support vector machines. Technical Report CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London, May, 1998.