Hacettepe Journal of Mathematics and Statistics
Volume 46 (3) (2017), 547 – 557

A single pairwise model for classification using online learning with kernels

Engin Tas *†

Abstract

Any binary or multi-class classification problem can be transformed into a pairwise prediction problem. This expands the data and brings an advantage of learning from a richer set of examples, in the expense of increasing costs when the data is in higher dimensions. Therefore, this study proposes to adopt an online support vector machine to work with pairs of examples. This modified algorithm is suitable for large data sets due to its online nature and it can also handle the sparsity structure existing in the data. Performances of the pairwise setting and the direct setting are compared in two problems from different domains. Results indicate that the pairwise setting outperforms the direct setting significantly. Furthermore, a general framework is designed to use this pairwise approach in a multi-class classification task. Result indicate that this single pairwise model achieved competitive classification rates even in large-scaled datasets with higher dimensionality.

Keywords: online learning, pairwise learning, support vector machines, kernel methods, multi-class classification.

2000 AMS Classification: 68W27, 62H30, 68T05, 68Q32

Received : 23.06.2016 *Accepted* : 02.08.2016 *Doi* : 10.15672/HJMS.2017.416

*Department of Statistics, Faculty of Science and Literature, Afyon Kocatepe University Email: engintas@aku.edu.tr
†Corresponding Author.

1. Introduction

In multi-class classification, we have a set of examples each belongs to one of the M different classes. The task is to learn a classification function which, given a new example, will predict the class to which the new observation belongs. Learning process starts by introducing a loss function which measures the error between the prediction and the actual class of the new example. An empirical risk function is then defined over a training data to estimate this loss accordingly. For instance, in classification tasks, we try to minimize the cost we pay for incorrectly assigning the examples to the wrong class.

There are two basic schemes in multi-class classification. The first one is one-versus-all (OVA) which we find M different binary classification functions, each one try to discriminate the examples in a single class from the examples in all remaining classes. For the ith classifier, we let the positive examples be all the observations in class i, and let the negative examples be all the observations not in class i. The second scheme, all-versus-all (AVA), builds $M(M-1)/2$ binary classifiers, one classifier for separating each pair of classes i and j. In both schemes, a new example is attained to the class which has the highest classifier score. Although, there exist so many studies proposing more complicated methods for multi-class classification, a recent study [27] showed that the OVA scheme is extremely powerful, using well-tuned binary classifiers such as support vector machines (SVMs), producing results that are often at least as accurate as other methods. In these approaches, the main problem lies in the fact that one has to find an effective way of combining different binary classifier functions in a single multi-class classification model.

The critical point in multi-class classification is to have a powerful binary classifier rather than a search for much more complex models. The central idea of SVM is to construct an optimal separating hyperplane over linearly separable data [5]. It can also learn a large margin hyperplane over linearly inseparable data by using kernels and soft margin formulations. SVMs also impose a form of regularization by controlling the capacity of the classifier function in order to avoid over-fitting or under-fitting problem. This provides a key support in building powerful binary classifiers. However, SVM is originally designed for binary classification and there are two principal approaches for extending SVM to the multi-class scenario. One approach is to generalize the binary algorithm to multi-class [25, 15], another approach is to decompose the multi-class problem into a series of binary problems. Pairwise classification is an alternative technique for solving multi-class problems by considering pairwise comparisons obtained from each of the two-class problems [10]. A test observation is assigned to the class that wins the most pairwise comparisons.

In this paper, we recast the multi-class classification problem in a pairwise setting. Let us denote a pair formed by two examples from the same class as a positive pair and a pair formed by two examples from different classes as a negative pair. Then, we construct a single pairwise SVM model for classifying pairs This pairwise SVM model is able to identify a given pair, whether it is a positive pair or not. However, in a pairwise setting n examples correspond to n^2 pairwise examples and training a support vector machine with such amount of data introduces serious computational costs in most cases with large scaled data. SVMs, in the batch setting, requires the evaluation of the objective function at each step and this involves basically the summation of a predefined loss over a set of data to be trained. Gradient-based methods must compute this sum for each evaluation of the objective, respectively its gradient whereas standard numerical optimization techniques such as variation of Newton's method, and the conjugate gradient algorithm needs the second-order information of the objective function. As available data

sets grow ever larger, such classical second-order methods are impractical in almost all useful cases.

Online gradient-based methods such as Perceptron [19, 16] and its variants [1, 9, 14, 11], by contrast, have a major advantage in large and redundant data sets. In fact, simple online gradient descent [6, 26, 21] outperforms sophisticated second-order batch methods in general since the computational requirements of online methods are extremely reduced by the fact that they only process examples one by one or small sub-samples of the training data. Therefore, this work depends on two basic ideas related to the problem setting and the choice of a proper method. First, we know that any binary or multi-class classification problem can be converted into a pairwise classification problem. This brings an advantage of learning from a larger data set of pairs. Secondly, traditional learning algorithms become inefficient in this setting and in some cases impossible to apply (e.g. batch methods), we propose to modify our single pairwise SVM algorithm to work with pairs of examples in an online manner. This brings an advantage of processing pairs one by one and overcomes the difficulties arising from large scaled data with higher dimensions.

2. General Framework

We have a set of examples $\mathcal{X} = (x_1, x_2, ..., x_m), \forall x_i \in \mathbb{R}^n$. One can think of any combination of two examples as a pair $p = (x_i, x_j) \in \mathcal{P} \subseteq \mathcal{X}^2$ and denote a sequence $T = \{(p, y_p) : p \in \mathcal{P}\}$ be a training sample drawn according to a probability distribution over a sample space $\mathbb{Z} = \mathcal{P} \times \{+1, -1\}$. An example from T is a triple consisting of a pair of an n-dimensional column vector of real-valued features and a corresponding label y_p indicating whether an interaction exists between these two pairs of examples. The task is to learn a suitable function $f : \mathcal{P} \to \{+1, -1\}$ from the training sample. We consider the linear case where the decision function represented as $f(p) = \langle w, \Phi(p) \rangle$, where $w \in \mathbb{R}^n$ is a vector of parameters that needs to be estimated based on training sample T.

An optimal decision function (optimal hyperplane with the biggest margin) is found by minimizing the following objective function in the feature space [20]:

$$\min_{w} \|w\|^2 + C \sum_{i=1}^{n} \xi_i \quad \text{with} \quad \begin{cases} \forall i & y_i \hat{y}(p_i) \geq 1 - \xi_i \\ \forall i & \xi_i \geq 0 \end{cases} \tag{2.1}$$

The slack variables $\xi = \xi_1, \xi_2, \ldots, \xi_n$ allow for some pairs to be on the wrong side of the margin. For very large values of the regularization parameter C, we have a high penalty for nonseparable examples and we may store many support vectors. Smaller values of C softens the effect of this penalty and produce better results on noisy problems but, we may have the risk of underfitting.

Maximizing the dual of this convex optimization problem is a simpler convex quadratic programming problem than the primal 2.1. The coefficients α_i of the SVM kernel expansion are found by defining the dual objective function.

$$W(\alpha) = \sum_i \alpha_i y_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j K(p_i, p_j) \tag{2.2}$$

and solving the SVM Quadratic Programming (QP) problem:

$$\max_{\alpha} W(\alpha) \quad \text{with} \quad \begin{cases} \sum_i \alpha_i = 0 \\ A_i \leq \alpha_i \leq B_i \\ A_i = \min(0, C y_i) \\ B_i = \max(0, C y_i). \end{cases} \tag{2.3}$$

Since we have a large number of growing pairs even in moderate size data sets, we need an effective SVM solver to handle these large data sets with large dimensions. A fast kernel classifier with online and active learning LASVM [‡] [4] is used to overcome these types of difficulties. Due to its online nature, it has several advantages in the pairwise setting. LASVM deals with complexity problems by processing examples one by one and keeping the most informative support vectors in its expansion. This reduces the amount of calculations significantly. Another advantage of LASVM is its compatibility for sparse data sets since the main problem in working with large dimensions, most of the data sets have a critical sparsity structure. LASVM overcomes this problem using convenient fast sparse vector products. This brings the benefits of learning from a richer data set formed by the combination of examples and by processing pairs of examples online. LASVM introduces a support vector removal step which means that the vectors collected in the current kernel expansion can be removed during the online process. It is also related to the sequential minimal optimization (SMO) [18] algorithm and converges to the solution of the SVM quadratic programming problem.

LASVM itself is not suitable for a pairwise setting. Because, in this form, you have to use its kernel cache to keep the kernel values between pairs and this brings quite expensive costs. This makes the algorithm unscalable to large data sets. It's also meaningless to keep kernel values between pairs, because once you have the kernel values between these examples than you need three operations to calculate the kernel value between any pair. Therefore, in this study, there are several modifications made at the implementation level in order to adapt LASVM to work in the pairwise setting. We name the resulting algorithm as pw-LASVM. The first main contribution of this paper is the following base modifications to LASVM in order to build pw-LASVM.

- We define the set P and S to keep the indices of the support pairs and the indices of the corresponding examples of the pairs respectively. When pw-LASVM inserts one pair to its current kernel expansion (process), we insert the index of the pair into the set P, and simultaneously add the corresponding indices of the two examples into the set S. We use the set P just to keep the indices of the pairs, it does not have a kernel cache since the kernel values are calculated for the examples as we mentioned before. On the other hand, the set S has a kernel cache which keeps the kernel values between examples.
- All the procedures for processing an example are modified to process a pair of examples. This includes the calculation of the gradient of a pair, identification of the τ-violating quadruple (formed by two pairs) with the maximal gradient and the direction searches.
- Reprocess removes some pairs from P. This corresponds to removing two examples of the related pair from the set S. Finally all the quantities like the bias term b and the gradient δ of the most τ-violating quadruple in the P are computed.

LASVM is an online kernel classifier introducing a support vector removal step which means that the vectors collected in the current kernel expansion can be removed during the online process. It is also related to the sequential minimal optimization (SMO) [18] algorithm and converges to the solution of the SVM quadratic programming problem. As LASVM is adapted to work with pairs of examples, it is named as pw-LASVM. It first inserts at least one pair into its current kernel expansion (process) and then searches for redundant support pairs that are already existed in the current kernel expansion (reprocess). In the online setting, this can be used to process a new pair at time t. Now, the pairs in the current kernel expansion with coefficient $\alpha_i \neq 0$ are called support pairs

[‡]http://leon.bottou.org/projects/lasvm

and the indices in the set S of potential support pairs correspond to the indices of pairs. The coefficient α_i are assumed to be null if $i \notin S$.

In a pairwise setting, the structure of pw-LASVM deserves much more attention, since the *process* and *reprocess* steps nicely retain more informative pairs and bail out the pairs that are not needed in the current kernel expansion. This is most useful in the pairwise setting where the pairs grow quadratically with the size of the data set.

2.1. Pairwise setting. Let's start with two introductory examples to explain the pairwise setting of the problem. First, assume a network formed by a set of proteins where one can observe several interactions between these proteins according to a particular biological function. A link is created between two proteins whenever an interaction occurs. Basically, the data we have from this event is the features of these proteins and a positive label indicating the interaction. This data can basically be structured in two ways:

- In the direct setting, two proteins are regarded as one example by combining their features and assigning a positive class membership. A sample of examples from negative class can also be generated in a similar way considering non-interacting proteins. If a new protein joins into this network, in order to predict new links between the new protein and existing ones, features of the new protein and the features of any existing protein in the network are combined as if a new example and compared with the examples which were previously formed and a label is assigned indicating the relation (interacting or non-interacting).
- In the indirect (pairwise) setting, we can think of two interacting proteins as a positive pair and two non-interacting proteins as a negative pair. There is no feature conjunction in this setup. Comparisons are made in a pairwise manner where all combinational relations considered between the members of the two pairs. In the first setup, the problem is defined as a standard binary classification task whereas in the second setting we have a link prediction problem between pairs of proteins. Consequently, the structure of the data we have also depends on the problem setup. Finally, this approach can also be applied in binary classification problems.

In order to learn a mapping from pairs, we need an additional type of structure for representing a pair in a joint feature space. [2] unified user ratings and item features in a common learning architecture and gave good examples of designing suitable kernels for different pairs of examples and also illustrated several ways of combining kernels into a single kernel. They used tensor products to simply join distinct feature maps. [17] proposed a kernel method for using combinations of features across example pairs in learning pairwise classifiers. In this context, [3] proposed the tensor product pairwise kernel (TPPK) that converts a kernel between single proteins into a kernel between pairs of proteins, and illustrated the kernel's effectiveness in conjunction with a support vector machine classifier. [23] proposed the metric learning pairwise kernel (MLPK) for the reconstruction of biological networks. [13] proposed a special case of this general framework where they used Cartesian kernel to speed up the online training process since, Cartesian kernel has a much sparser structure than TPPK and MLPK. They also provide generalization bounds for the two pairwise kernels based on eigenvalue analysis of the kernel matrices.

In this paper, we used TPPK in the conjunction with an element-wise kernel between examples. TPPK between the pairs $p_1 = (x_1, x_2)$ and $p_2 = (x_3, x_4)$ is given as

$$K_{\text{TPPK}}((x_1, x_2), (x_3, x_4)) = K(x_1, x_3)K(x_2, x_4) + K(x_1, x_4)K(x_2, x_3). \tag{2.4}$$

With an appropriate choice for element-wise kernel, such as the Gaussian RBF kernel, the kernel TPPK generates a class $\mathcal{H}$ of universally approximating functions for learning

any type of relation between pairs. We had also examined the performance of MLPK in all experiments and didn't find any statistically significant differences from TPPK.

3. Experiments

In the first part of the experiments, we demonstrated the performance of the pairwise setting with respect to the direct setting in two examples from different domains. These examples have distinct characteristics in dimensionality and sample size. Second part of the experiments demonstrates the useful application of the pairwise model on the standard multi-class or binary classification problems. In all experiments, pw-LASVM is used with a soft margin loss and a small tolerance $\tau = 0.001$ on gradients. Unless otherwise specified, a grid-search is done for the regularization parameter C in the interval $[0.001, 0.1, 1, 10, 100, 100]$ and for the γ parameter of the Gaussian RBF kernel in the interval $[0.001, 0.01, 1, 10]$.

3.1. Pairwise Experiments. In the first example, we have two biological networks, a protein-protein interaction network (von-Mering) [24] and a metabolic network (ligand)[§] [22]. A biological network can be regarded as a graph with proteins as nodes and protein-protein relations as edges. In each network, an edge indicates that the two proteins are enzymes that catalyze successive reactions between them according to a particular function. The protein-protein interaction network contains 2617 nodes and 11855 edges. Each protein is described by a 76-dimensional feature vector, where each feature indicates whether the protein is related to a particular function or not. The metabolic network contains 755 nodes and 7860 edges where each protein is described by a 36-dimensional feature vector. Throughout the text, we denote an interacting pair as a positive pair and non-interacting pair as a negative pair. We start with forming a positive pairs set by obtaining all interacting pairs from the network. Then we generate all negative pairs set from the remaining non-interacting proteins. In order to have a balanced data set we include all positive pairs and sample randomly from negative pairs set with the sample size equals to the number of positive pairs. These sets of positive and negative pairs are then combined to form training pairs. This training pairs data is used to create 3×5 cross-validation data sets for the first set of experiments. Given a new pair of proteins, the task is to determine whether it's a positive (interacting) pair or not.

Linear and Gaussian RBF kernels are used as element-wise kernels in conjunction with the pairwise kernels. A grid search is done for the regularization parameter C and the parameter γ of the RBF kernel. Best parameter pairs obtained from the grid-search are given in Table 1. Kernel cache size is set to 256MB.

We compared the classification performance of the pairwise setting and the direct setting according to the classification accuracy (ACC) and the area under the ROC curve (AUC). Table 2 summarizes results from 3×5 cross validation. We see that the pairwise settings achieved significantly better classification performances than the direct setting.

In the second example, we used 20-Newsgroups [¶] data set which is a collection of approximately 20000 documents organized in 20 distinct newsgroups, each corresponding to a different topic. Each document is described by a 25736-dimensional vector formed by frequencies of words occurring in the document. We formed the training pair data set using documents from two groups related to sports and computers. Two documents in the same group are considered as a positive pair and as a negative pair if they are in different groups. Since the number of pairs grows quadratically with the number of documents.

[§]http://www.genome.ad.jp/ligand

[¶]http://qwone.com/~jason/20Newsgroups/

Table 1. Best parameter values obtained from the grid-search

Data set	Setting	Element-wise kernel	Pairwise kernel	C	γ
Ligand	Direct	Linear	-	1	-
		RBF	-	1	1
	Pairwise	Linear	TPPK	1	-
		RBF	TPPK	10	0.1
von-Mering	Direct	Linear	-	1	-
		RBF	-	1	1
	Pairwise	Linear	TPPK	0.1	-
		RBF	TPPK	10	0.1

[a] C and γ is chosen from $[0.01, 0.1, 1, 10, 100]$

Table 2. Classification performance comparisons between the direct and indirect approaches with linear and Gaussian RBF element-wise kernels

	Linear		RBF	
	Direct	TPPK	Direct	TPPK
Ligand	0.65 / 0.70	**0.77 / 0.82**	0.73 / 0.76	**0.82 / 0.87**
von-Mering	0.68 / 0.73	**0.79 / 0.86**	0.78 / 0.85	**0.84 / 0.90**

[a] Bold numbers indicate statistically significant differences from the direct approach.

For 10000 documents, we have 100 million pairs of documents in total. Therefore, it is impossible to include all the positive pairs existing in the data; a balanced data set is formed by randomly sampling from the whole set of positive and negative pairs with sample sizes ranging from 10000 to 250000. Test pairs data set are created in the same manner as in the training, but using different newsgroups rather than the groups used in the training in order to have independent training and test sets. Given a pair of documents, the task is to determine whether these documents are in the same group or not.

At first, we tried to use linear and Gaussian RBF kernels as element-wise kernels, but we observed that the RBF kernel performs poor in this data set because of the curse of the higher dimensionality. This conforms with the fact that RBF kernel has lost the sense of locality [12, 8]. Therefore, we used the linear kernel as an element-wise kernel. A grid search is done for the regularization parameter C, and we found $C = 10$ as the optimal value. Kernel cache size is set to 512MB. Since this data set are not normalized beforehand, all kernel values between examples are normalized in an online manner during the calculation of the pairwise kernel.

In this experiment, we tried to see the effect of enriching the original data by forming pairs using several combinations of documents. Therefore, we generated training pair data sets with sample sizes ranging from 10000 to 250000. Results indicated that indirect setting performs significantly better than the direct setting. Figure 1 shows that the classification performances of the pairwise setting are increased proportionally with the size of the training data whereas the direct setting performs poor almost in all cases.

3.2. Pairwise to Multi-class Classification. This section is the second main contribution of the paper which we would like to demonstrate the idea of using a pairwise

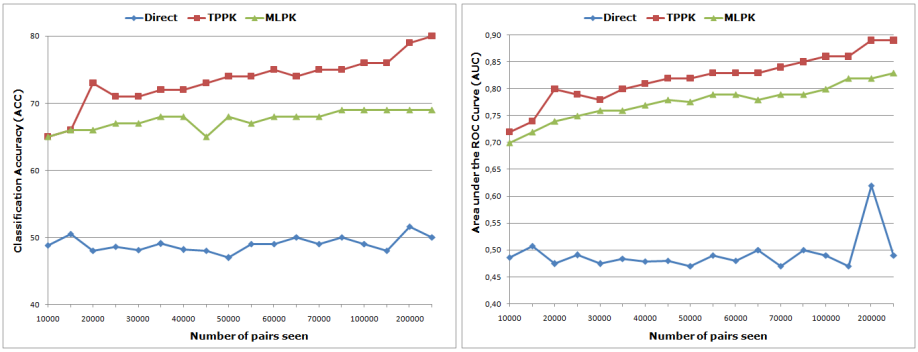

Figure 1. Classification performances in 20-Newsgroups data set using criteria a)ACC b)AUC

SVM model in a multi-class classification task. Let us start with an introductory example which is a simple character recognition task. First, assume that each example in the data set represents a character written in a specific font. We coded these characters with a letter and a number (A1, A2, B1, B3, C1, C2, etc.) in 2. We used A1 and A2 to denote the binary images of handwritten letter "A" which are written in different fonts. The task is to recognize these characters and assign the correct label (A,B,C, etc.).

We can think of the whole process in two stages such as the training and the testing. The first step is to preprocess the data in order to make it convenient for pairwise learning. Therefore, the data is divided into three sets such as train, validation and test. Then, pairwise train data is formed by taking all pairwise combinations (if possible, otherwise we randomly sample from all pairwise combinations) in the train set and a pairwise validation data is formed similarly by taking all possible pairwise combinations in the validation set. These pairwise data sets are formed by positive and negative pairs of examples, where we assign a positive label if it includes two examples from the same class and a negative label if it includes examples form different classes. An online pairwise SVM model is trained on pairwise train data. Pairwise classification performance of the trained SVM model is evaluated on the pairwise validation data and the hyper-parameters of the model is tuned. Once an ideal model is built, we proceed to the testing stage.

In the testing stage, consider a test example which we have never seen before, we couple it with a specific number (ex. 100) of train examples. Proceeding in this way, by coupling test examples with a specified number of train examples (with known labels), we form the pairwise test data. This pairwise test data is given as an input to the built pairwise SVM model and the pairwise predictions obtained. Once we have these predictions, we have the votes of the specified number of train examples for the corresponding test example. This provides us with the information whether the test example is coming from the same class of the corresponding training example or not. In other words, every train example in a pair gives a vote to the test example whether it's coming from the same class or not. Finally, we simply use the majority vote to assign the test example to the winning class. The whole process is illustrated in Figure 2.

For the final experiments, 20-Newsgroups and MNIST [‖] datasets are used for multi-class classification. 20-Newsgroups datasets is used again in the final experiments in order to demonstrate the scalability of the pw-LASVM and the above approach. Up to

[‖]http://yann.lecun.com/exdb/mnist

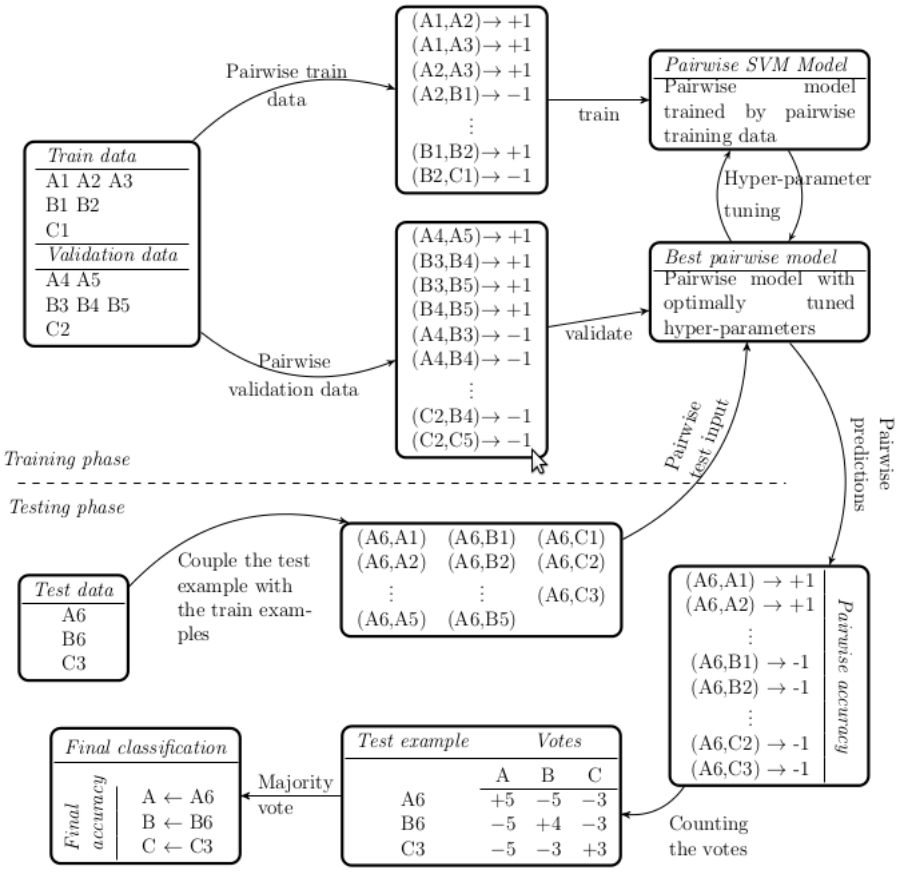

Figure 2. Schematic illustration of multi-class classification using pw-LASVM

Table 3. Pairwise and final classification accuracies of best models trained according to the given process in figure 2

		Nr. of training pairs	Nr. of voters	Pairwise accuracy	Final accuracy
Data set	20-Newsgroups	250000	200	90.09	83.00
	MNIST	100000	100	95.89	94.00

now, multi-class classification experiments are limited to small-scaled datasets both in sample size and dimensionality. MNIST dataset is chosen in order to see the performance of the new approach in a problem with low dimensionality.

In 20-Newsgroups experiments, linear kernel is used as element-wise kernel, pw-LASVM is initialized with a regularization parameter $C = 10$. Kernel cache size is set to 1024MB. Since this data set are not normalized beforehand, all kernel values between examples are normalized in an online manner during the calculation of the pairwise kernel. In

MNIST experiments, Gaussian RBF kernel is used with $\gamma = 0.01$ as element-wise kernel, $C = 1000$ is set for the regularization parameter.

Results of the final experiments are summarized in Table 3. Although, these are not the best results when comparing with the existing methods in the literature, it's noteworthy that pw-LASVM achieved these classification performances using a single pairwise model which is totally different from general approaches such as "one-against-all" and "one-against-one".

4. Conclusion

pw-LASVM is developed by effectively modifying the LASVM algorithm as to work with pairs of examples. Performance of the pw-LASVM is demonstrated in two types of relational networks. Results demonstrated that, the pairwise approach achieved a better performance than the direct approach. A general framework is built in order to solve multi-class classification problems with pw-LASVM. Final experiments indicate that the proposed approach is scalable to large datasets with higher dimensionality.

pw-LASVM can be used in binary, multi-class classification problems in domains such as statistics, finance, information retrieval,collaborative filtering and social network analysis where the data is very large with higher dimensionality. On the other hand, in domains like bioinformatics, genetics and biostatistics, since the cost of obtaining a sample is quite high, we have data sets with small sample sizes in classification or link prediction problems. This makes the model building process much harder because there is not enough data to validate or even test the model performance. In these cases, this approach can also be applied since one can extend this data by transforming the classification problem into a pairwise setting, and use pw-LASVM as an effective online SVM solver which can work with pairs of examples using suitable pairwise kernels.

References

[1] Anlauf, J. and Biehl, M. (2007). The adatron: an adaptive perceptron algorithm. *EPL (Europhysics Letters)*, **10**(7), 687.

[2] Basilico, J. and Hofmann, T. (2004). Unifying collaborative and content-based filtering. In *Proceedings of the twenty-first international conference on Machine learning*, ICML '04, pages 9–, New York, NY, USA, 2004. ACM. ISBN 1-58113-838-5. doi: 10.1145/1015330.1015394. URL `http://doi.acm.org/10.1145/1015330.1015394`.

[3] Ben-Hur, A. and Noble, W. (2005). Kernel methods for predicting protein-protein interactions. *Bioinformatics*, **21**(suppl 1), i38–i46.

[4] Bordes, A., Ertekin, S., Weston, J., and Bottou, L. (2005). Fast kernel classifiers with online and active learning. *Journal of Machine Learning Research*, **6**, 1579–1619.

[5] Boser, B., Guyon, I., and Vapnik, V. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM.

[6] Bottou, L. and LeCun, Y. (2004). Large scale online learning. In Thrun, S., Saul, L., and Schölkopf, B., editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA. URL `http://leon.bottou.org/papers/bottou-lecun-2004`.

[7] Dietterich, T. and Bakiri, G. (1995). Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, **2**(263), 286.

[8] Francois, D., Wertz, V., and Verleysen, M. (2005). About the locality of kernels in high-dimensional spaces. In *Proceedings of ASMDA 2005, International Symposium on Applied Stochastic Models and Data Analysis*, pages 238–245. URL `http://hdl.handle.net/2078.1/93830`.

[9] Freund, Y. and Schapire, R. (1999). Large margin classification using the perceptron algorithm. *Machine learning*, **37**(3), 277–296.

[10] Friedman, J. (1996). Another approach to polychotomous classifcation. Technical report, Technical report, Stanford University, Department of Statistics.

[11] Gentile, C. (2002). A new approximate maximal margin classification algorithm. *J. Mach. Learn. Res.*, **2**, 213–242. ISSN 1532-4435. URL `http://portal.acm.org/citation.cfm?id=944790.944811`.

[12] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2 edition. ISBN 0387927091. URL `http://www-stat.stanford.edu/~tibs/ElemStatLearn/`.

[13] Kashima, H., Oyama, S., Yamanishi, Y., and Tsuda, K. (2009). On Pairwise Kernels: An Efficient Alternative and Generalization Analysis. In *Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, PAKDD '09, pages 1030–1037, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 978-3-642-01306-5. URL `http://dx.doi.org/10.1007/978-3-642-01307-2\_110`.

[14] Li, Y. and Long, P. (2002). The relaxed online maximum margin algorithm. *Machine Learning*, **46**(1), 361–387.

[15] Mayoraz, E. and Alpaydin, E. (1999). Support vector machines for multi-class classification. *Engineering Applications of Bio-Inspired Artificial Neural Networks*, pages 833–842.

[16] Minsky, M. and Seymour, P. (1969). Perceptrons.

[17] Oyama, S. and Manning, C. D. (2004). Using feature conjunctions across examples for learning pairwise classifiers. In *15th European Conference on Machine Learning (ECML2004)*. URL `http://ilpubs.stanford.edu:8090/767/`.

[18] Platt, J. C. (1999). 12 fast training of support vector machines using sequential minimal optimization.

[19] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, **65**(6), 386.

[20] Schölkopf, B. and Smola, A. (2001). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT press.

[21] Shalev-Shwartz, S., Singer, Y., and Srebro, N. (2007). Pegasos: Primal estimated sub-gradient solver for svm. In *Proceedings of the 24th international conference on Machine learning*, pages 807–814. ACM.

[22] Vert, J.-P. and Kanehisa, M. (2003). Graph-driven features extraction from microarray data using diffusion kernels and kernel cca. *Advances in Neural Information Processing System*, **5**, 1425–1432.

[23] Vert, J.-P., Qiu, J., and Noble, W. (2007). A new pairwise kernel for biological network inference with support vector machines. *BMC Bioinformatics*, **8**(Suppl 10), S8.

[24] von Mering, C., Krause, R., Snel, B., Cornell, M., Oliver, S. G., Fields, S., and Bork, P. (2002). Comparative assessment of large-scale data sets of protein–protein interactions. *Nature*, **417**(6887), 399–403.

[25] Weston, J. and Watkins, C. (1999). Support vector machines for multi-class pattern recognition. In *Proceedings of the seventh European symposium on artificial neural networks*, volume 4, pages 219–224.

[26] Xu, W. (2011). Towards optimal one pass large scale learning with averaged stochastic gradient descent. *arXiv preprint arXiv:1107.2490*.

[27] Rifkin, R. and Klautau, A. (2004). In defense of one-vs-all classification. *Journal of machine learning research*, volume 5, pages 101–141.

Appendices

The codes developed and the datas used in this paper can be found under the link `http://blog.aku.edu.tr/engintas/publications/`