# Distance Function Learning in Error-Correcting Output Coding Framework

Dijun Luo and Rong Xiong

National Lab of Industrial Control Technology, Zhejiang University, China

**Abstract.** This paper presents a novel framework of error-correcting output coding (ECOC) addressing the problem of multi-class classification. By weighting the output space of each base classifier which is trained independently, the distance function of decoding is adapted so that the samples are more discriminative. A criterion generated over the Extended Pair Samples (EPS) is proposed to train the weights of output space. Some properties still hold in the new framework: any classifier, as well as distance function, is still applicable. We first conduct empirical studies on UCI datasets to verify the presented framework with four frequently used coding matrixes and then apply it in RoboCup domain to enhance the performance of agent control. Experimental results show that our supervised learned decoding scheme improves the accuracy of classification significantly and betters the ball control of agents in a soccer game after learning from experience.

## 1 Introduction

Many supervised machine learning tasks can be cast as the problem of assigning patterns to a finite set of classes, which is often referred to as multi-class classification. Examples include optical character recognition (OCR) system addresses the problem of determining the digit value of an image, text classification, speech recognition, medical analysis, and situation determination in robot control etc.. Some of the well known binary classification learning algorithms can be extended to handle multi-class problems [4, 16, 17]. Recently it becomes a general approach to combine a set of binary classifiers to solve a multi-class problem.

Dietterich and Bakiri [7] presented a typical framework of this approach, which is known as error-correcting output coding (ECOC), or output coding in short. The idea of ECOC enjoys a significant improvement in many empirical experiments [7, 8, 1, 18, 3, 2].

The methods of ECOC previously discussed, however, are based on a predefined output code and a fixed distance function. In this case, a predefined code is used to encode the base learners, and the predefined output code and a distance function is employed to compute the discriminative function, according to which a testing instance is assigned to some class. Crammer and Singer argued that the complexity of the induced binary problems would be ignored due to the predefinition of the output code. Hence a learning approach of designing an output code is presented [5].

This paper illustrates another way of adapting the decoding process of ECOC framework by learning approach which yields a significant improvement of multi-class classification in several empirical experiments. The major idea is redefining the distance

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 1–10, 2006.

function by rescaling the output space of every base learner which is trained independently. By employing the idea of Vapnik's support vector machines (SVMs) we define a criteria as the sum of empirical hinge loss and the regularization with a trade-off factor between them. The criteria is generated over the *Extended Pair Samples (EPS)* which contain a subset of pair-instances as ranking SVMs.

Two experiments are conducted for validation of the performance of our method. The first is on UIC Repository and the second is on RoboCup domain. The experimental results show that our method outperforms the existing approaches significantly.

## 2 ECOC Framework

In ECOC framework, all base classifiers are trained independently. This training scheme ignores the dataset distribution and the performance of each base classifier. Though some probability based decoding methods are introduced in [14], the following problem remains unsolved: the criterion of a good is not well defined. Therefore, what is a better or best decoding function is not clear. In this paper, we illustrate a clear scheme of defining an optimal decoding function. The method proposed in this paper is different from finding an optimal decoding matrix which is first used by Crammer & Singer [5], and is probably much more efficient, because the optimization space is much simpler than that used in Crammer & Singer's method.

### 2.1 Scheme of Error-Correcting Output Coding

Let $S = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$ denotes a set of training data where each instance $x_i$ belongs to a domain $\mathcal{X}$ and each label $y_i$ belongs to a set of labels representing categories $\mathcal{Y} = \{1, 2, ..., k\}$, and $N$ is the number of instances. A multi-class classifier $H : \mathcal{X} \mapsto \mathcal{Y}$ is a function that maps an instance $x$ in $\mathcal{X}$ into a label $y$ in $\mathcal{Y}$.

A typical ECOC method is conducted as follows,

(1) **Encoding:** A codeword $M$ is defined. $M$ is a matrix of $k \times n$ size over $\{-1, 0, +1\}$ where $k$ is size of label $\mathcal{Y}$ set and $n$ is number of binary classifiers. Each row of $M$ correspond to a category and each column corresponds to a binary classifier. The $n$ binary base classifiers are denoted as $h_1(x), h_2(x), ..., h_n(x)$.

Several families of codes have been proposed and tested so far for encoding, such as, comparing each category against the rest , comparing all pairs of categories (one-against-one), employing the random code, and employing the Hadamard code [7, 9, 11].

(2) **Base classifier construction:** A dichotomy of samples is created for each base classifier. The dichotomies vary according to classifiers. If $M_{y,s} = -1$, we take all the instances labeled $y$ as negative samples in training set of the base learner $h_s$ . If $M_{y,s} = 1$, we take all the instances labeled $y$ as positive samples in training set of the base learner $h_s$. If $M_{y,s} = 0$, the instances are ignored. SVMs can be used as the model of base classifier.

(3) **Decoding:** Given an instance $x$, a vector of binary labels is generated from all the base classifiers $\mathcal{H}(x) = (h_1(x), h_2(x), ..., h_n(x))$. We then compare the vector with each row of the matrix $M$ (each category). A final classification decision is made using the discriminate function as follows,

$$H(x) = \arg\min_{y \in \mathcal{Y}} \mathcal{F}(x, y) \tag{1}$$

$$\mathcal{F}(x, y) = D(M_y, \mathcal{H}(x)) \tag{2}$$

where $D(u, v)$ is distance function between vectors $u$ and $v$, and $M_y$ is the row $y$ of the code matrix $M$. Consequently, the label of $x$ is predicted to be $y$ if the output of base classifiers is the 'closest' to the row of $M_y$.

### 2.2 ECOC Framework with Decoding Learning

A lot of empirical experiments show that ECOC enjoys a significant improvement [7, 8, 1, 18, 3, 2]. One, however, argues that ECOC suffers the following problem [15]: Hamming decoding scheme ignores the confidence of each classifier in ECOC and this confidence is merely a relative quantity which means using a linear loss base distance function in decoding may introduce some bias in the final classification in the sense that classifiers with a larger output range will receive a higher weight. Thus both Hamming distance function and simple loss base distance function have disadvantage. In [15] a probability based decoding distance function is proposed. The relation between an optimal criterion and the parameters of the distance function is not well defined. Therefore in fact, the introduction of probability based distance function is just an approximation of an optimal decoding. This paper presents a learning approach to searching an optimal distance function for ECOC decoding which will overcome the problem suffered by previous work.

## 3 Distance Function Learning

In this paper we present a novel algorithm of multi-class classification (which is termed OC.MM) by introducing the max margin distance function learning in ECOC.

We rewrite the distance function as the following form,

$$D(u, v) = \sum_{s=1}^{n} d(u_s, v_s).$$

which implies that the distance or similarity is composed of each dimension independently. This property holds in most of the existing distance function include hamming distance and linear distance. In our distance learning approach, we assign each dimension of the output of base learner a weight, so that the output space of $\mathcal{H}(x)$ is rescaled. The larger the distance is, the less the similarity is. Thus we can equivalently consider a weighted version of similarity function as,

$$K(u, v) = \sum_{s=1}^{n} w_s k(u_s, v_s).$$

Consequently the final classification hypothesis is

$$y = H(x) = \arg\max_{y} \left( \sum_{s=1}^{n} w_s k(M_{y,s}, h_i(x)) \right) \tag{3}$$

We denote

$$F(x, y; w) = \sum_{s=1}^{n} w_s k(M_{y,s}, h_i(x)) = \langle w, \sigma_y(x) \rangle, \tag{4}$$

where $w = [w_1, w_2, ...w_n]$,$\sigma_y = [k(M_{y,1}, h_1(x)), k(M_{y,2}, h_2(x)), ...k(M_{y,n}, h_n(x))]$, and $\langle u, v \rangle$ denotes the inner product of $u$ and $v$.

In order to illustrate our method of max margin decoding distance function, we first define the Extended Pair Samples (EPS) as follows,

$$S^{EPS} = \left\{ \left( [\sigma_{y_k}(x_i), \sigma_{y_j}(x_i)], z_{i,y_k,y_j} = \begin{cases} 1, y_k = y_i, y_j \neq y_i \\ -1, y_j = y_i, y_k \neq y_i \end{cases} \right) : (x_i, y_i) \in S \right\}. \tag{5}$$

### 3.1 Primal QP Problem and Dual Problem

We consider the multi-class classification problem as a ranking one. An instance is correctly classified if a pattern $\sigma_{y_i}(x_i)$ ranks first in a subset of $S^{EPS}$ given any instance $x_i$. That is

$$F(x_i, y_i; w) \geq F(x_i, y; w), \forall y \in \mathcal{Y}, y \neq y_i. \tag{6}$$

Then the criteria of OC.MM is as follows,

$$\min_{w} \sum_{\omega \in S^{EPS}} \left[ 1 - \langle w, \sigma_{y_j}(x_i) - \sigma_{y_k}(x_i) \rangle \right]_+ + \lambda \|w\|^2 \tag{7}$$

where $\omega = \left( [\sigma_{y_k}(x_i), \sigma_{y_j}(x_i)], z_{i,y_k,y_j} \right)$, $[z]_+ = \max(0, z)$ and $\lambda$ is a wight between the regularization and the hinge loss. Instead of solving the above optimization, we solve the following equivalent one [10],

$$\frac{1}{2} \min_{w} + C \sum_{\omega \in S^{EPS}} \xi_\omega \tag{8}$$

*s.t.*

$z_{i,y_j,y_k} \langle w, \sigma_{y_j}(x_i) - \sigma_{y_k}(x_i) \rangle \geq 1 - \xi_\omega, \xi_\omega \geq 0.$

Employing the Lagrangian multiplier method, the Lagrange function of (8) can be written as,

$$\mathcal{L}(w, \alpha, \xi, \zeta) = \frac{1}{2} \min_{w} + C \sum_{\omega \in S^{EPS}} \xi_\omega - \sum_{\omega \in S^{EPS}} \zeta_\omega \xi_\omega$$

$$- \sum_{\omega \in S^{EPS}} \alpha_\omega \left( z_{i,y_j,y_k} \langle w, \sigma_{y_j}(x_i) - \sigma_{y_k}(x_i) \rangle + 1 - \xi_\omega \right). \tag{9}$$

According to KKT conditions,

$$\frac{\partial \mathcal{L}_D}{\partial w_s} = 0 \Longleftrightarrow w_s = \sum_{\omega \in S^{EPS}} \alpha_\omega z_{i,y_j,y_k} \left( \sigma_{y_j}(x_i) - \sigma_{y_k}(x_i) \right) \tag{10}$$

$$\frac{\partial \mathcal{L}_D}{\partial \xi_\omega} = 0 \Longleftrightarrow C - \alpha_\omega - \zeta_\omega = 0 \tag{11}$$

Since $\zeta_\omega > 0$, optimization problem (8) reduces to a box constraint $0 \leq \alpha_\omega \leq C$. By substituting (10) and (11) into (9), we obtain the Lagrangian dual objective (12),

$$\mathcal{L}_D(\alpha) = \sum_{\omega \in S^{EPS}} \alpha_\omega -$$

$$\frac{1}{2} \sum_{\omega \in S^{EPS}} \sum_{\omega' \in S^{EPS}} \alpha_\omega \alpha_{\omega'} z_{i,y_j,y_k} z_{i',y'_k,y'_j} \langle \sigma_{y_j}(x_i) - \sigma_{y_k}(x_i), \sigma_{y'_j}(x_{i'}) - \sigma_{y'_k}(x_{i'}) \rangle, \tag{12}$$

where $\omega = \left( [\sigma_{y_k}(x_i), \sigma_{y_j}(x_i)], z_{i,y_k,y_j} \right)$ and $\omega' = \left( [\sigma_{y'_k}(x_{i'}), \sigma_{y'_j}(x_{i'})], z_{i',y'_k,y'_j} \right)$.

The solution of the dual QP is thus characterized by

$$\max_\alpha \mathcal{L}_D(\alpha)$$

*s.t.*

$$0 \leq \alpha_\omega \leq C, \forall \omega = \left( [\sigma_{y_k}(x_i), \sigma_{y_j}(x_i)], z_{i,y_k,y_j} \right) \in S^{EPS} \tag{13}$$

We notice that it is easy to generalize the linear learning algorithm to non-linear cases using kernel functions. Substituting (10) into (4), the following is derived,

$$F(x, y, w) = \sum_{\omega \in S^{EPS}} \alpha_\omega z_{i,y_j,y_k} \langle \sigma_{y_j}(x_i) - \sigma_{y_k}(x_i), \sigma_y(x) \rangle. \tag{14}$$

Replacing the inner products $\langle \sigma_{y_j}(x_i) - \sigma_{y_k}(x_i), \sigma_y(x) \rangle$ and $\langle \sigma_{y_j}(x_i) - \sigma_{y_k}(x_i), \sigma_{y'_j}(x_{i'}) - \sigma_{y'_k}(x_{i'}) \rangle$ with $K\left( \sigma_{y_j}(x_i) - \sigma_{y_k}(x_i), \sigma_y(x) \right)$ and $K\left( \sigma_{y_j}(x_i) - \sigma_{y_k}(x_i), \sigma_{y'_j}(x_{i'}) - \sigma_{y'_k}(x_{i'}) \right)$, where $K(u, v)$ is a kernel function, one can make the generalization. Then we obtain a nonlinear weighted decoding distance optimization criterion of algorithm OC.MM as follows,

$$\mathcal{L}_D(\alpha) = c^T \alpha - \alpha^T \Lambda \alpha \tag{15}$$

where $\Lambda$ is the kernel matrix containing all the kernel values over $S^{EPS}$ and $c = [1, 1, ...1]$. The final classification hypothesis as following,

$$y = \arg\max_y F(x, y, w) = \sum_{\omega \in S^{EPS}} \alpha_\omega z_{i,y_j,y_k} K\left( \sigma_{y_j}(x_i) - \sigma_{y_k}(x_i), \sigma_y(x) \right). \tag{16}$$

### 3.2 Effective Training Scheme

To faster the convergence of the algorithm above we introduce an effective training scheme which is shown in Algorithm 1.

The algorithm above is implemented by modifying Joachims' $SVM^{light}$ [12].

**Algorithm 1.** Effective algorithm for solving OC.MM

Input:$S^{EPS}$, $C$, $\epsilon$, $p$
$S_i \leftarrow \Phi, i = 1, 2, ..., N$
Randomly choose instances from $S^{EPS}$ into $S_i$ with probability $p$.

**repeat**
  **for all** $i$ such that $0 \le i \le N$ **do**
    $Q(y) = 1 - \sum_{\omega \in S^{EPS}} \alpha_\omega z_{i,y_j,y_k} K\left(\sigma_{y_j}(x_i) - \sigma_{y_k}(x_i), \sigma_y(x)\right)$
    $\hat{y}_i = \arg\max_{y \in \mathcal{Y}} Q(y)$
    $\hat{Q} = Q(\hat{y})$
    $\xi_i = \left[\max_{y \in S_i} Q(y)\right]_+$
    **if** $\hat{Q} > \xi_i + \epsilon$ **then**
      $S_i \leftarrow S_i \cup \left([\sigma_{y_i}(x_i), \sigma_{\hat{y}}(x_i)], z_{i,y_i,\hat{y}}\right) \cup \left([\sigma_{\hat{y}}(x_i), \sigma_{y_i}(x_i)], z_{i,\hat{y},y_i}\right)$
      $\alpha_{S_w} \leftarrow$optimize dual over $S_w = \cup_i S_i$
    **end if**
  **end for**
**until** $S_w$ dose not change.

## 4 Evaluations

Two experiments are conducted to evaluate the performance of the approach of OC.MM proposed in this paper. The first is conducted on 10 datasets selected from the UCI Repository. The second test-bed from the study on the application of our method in the domain of agent control.

### 4.1 Experimental Result on UCI Repository

We choose 11 datasets on UCI Repository to conduct this experiment. The datasets statistics are given in Table 1.

Four frequently used coding matrixes are applied in the experiments: one vs one, one vs rest, Hadamard, and random. In each we run $SVM^{light}$ [12] as the baseline. We set the random code to have 2k columns for the problem which has k classes. The entry in

**Table 1.** Statistics on UCI datasets

| Problem | #train | #test | #Attribute | #class |
|---|---|---|---|---|
| Glass | 214 | 0 | 9 | 6 |
| Segment | 2310 | 0 | 19 | 7 |
| Pendigits | 7494 | 3498 | 16 | 10 |
| Yeast | 1484 | 0 | 8 | 10 |
| Vowel | 528 | 0 | 10 | 11 |
| Shuttle | 43500 | 14500 | 9 | 7 |
| Soybean | 307 | 376 | 35 | 19 |
| Wine | 178 | 0 | 13 | 3 |
| Dermatology | 366 | 0 | 34 | 6 |
| Vehicle | 846 | 0 | 18 | 4 |

matrix is set to be -1 or +1 uniformly at random. Hadamard code is generated by the following scheme,

$$H_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, H_{n+1} = \begin{pmatrix} H_n & H_n \\ H_n & -H_n \end{pmatrix}.$$

For the base line we chose SVMs with the RBF kernels $K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}$ as the base classifiers. We tune the cost parameters $C$ in set $C = [2^{-6}, 2^{-5}, ..., 2^8]$ and $\gamma$ from set $\gamma = [2^{-10}, 2^{-9}, ..., 2^4]$, and choose the best result for each algorithm.For the datasets in which the number of training instances is less than 2000 or there are no testing data, we use a 10-fold cross validation.

EPS of our algorithm is generated from the same output of SVM$^{light}$. Thus the accuracy of SVM$^{light}$ is that of OC.MM without learning and with equal weights. Experimental results are shown in Table 2 from which we can see a significant improvement after applying our algorithm. Out of the $11 \times 4 = 44$ results, OC.MM outperforms SVM$^{light}$ in 35; they draw in the rest.

**Table 2.** Prediction accuracy of SVM$^{light}$ (SVM) and OC.MM on UCI datasets

| | *One-vs-one* | | *One-vs-rest* | | *Random* | | *Hadamard* | |
|---|---|---|---|---|---|---|---|---|
| Problem | SVM | OC.MM | SVM | OC.MM | SVM | OC.MM | SVM | OC.MM |
| Satimage | 0.9204 | 0.9204 | 0.8933 | 0.8979 | 0.9176 | 0.9191 | 0.9159 | 0.9182 |
| Glass | 0.6728 | 0.6962 | 0.6822 | 0.6962 | 0.7009 | 0.7009 | 0.6822 | 0.7056 |
| segmentation | 0.9718 | 0.9735 | 0.9528 | 0.9640 | 0.9606 | 0.9671 | 0.9606 | 0.9645 |
| Pendigits | 0.9958 | 0.9958 | 0.9940 | 0.9958 | 0.9952 | 0.9952 | 0.9950 | 0.9952 |
| Yeast | 0.5923 | 0.5923 | 0.4791 | 0.4791 | 0.5404 | 0.5606 | 0.4696 | 0.4716 |
| Vowel | 0.9886 | 0.9886 | 0.9772 | 0.9791 | 0.9753 | 0.9829 | 0.9753 | 0.9772 |
| Shuttle | 0.9970 | 0.9970 | 0.9969 | 0.9971 | 0.9971 | 0.9972 | 0.9971 | 0.9972 |
| Soybean | 0.9414 | 0.9428 | 0.9136 | 0.9341 | 0.9443 | 0.9487 | 0.9428 | 0.9502 |
| Wine | 0.9490 | 0.9490 | 0.9157 | 0.9550 | 0.9157 | 0.9550 | 0.9325 | 0.9438 |
| Dermatology | 0.9726 | 0.9754 | 0.9480 | 0.9644 | 0.9672 | 0.9726 | 0.9453 | 0.9754 |
| Vehicle | 0.8475 | 0.8475 | 0.8392 | 0.8534 | 0.8498 | 0.8747 | 0.8333 | 0.8546 |

### 4.2 Empirical Study on Agent Control

We conduct the second experiment on the task of opponent action prediction to evaluate the effectiveness of our algorithm. The test-bed is RoboCup robot soccer simulation which offers a special type of benchmark requiring real-time sensor evaluation and decision making, acting in highly dynamic and competitive environment etc. [13]. In this paper we focus on the task of predicting the action of an opponent possessing the ball in such an environment. This is an important subtask in RoboCup soccer game which enables our agents to model the opponents' action pattern. For example, when our agents are defending in front of our goal, it is more like to disorganize the opponent's attack if the agents could accurately predict who will the opponent possessing the ball

will pass to. The prediction is viewed as a multi-class classification problem on the target space as follows,

$$A = \{pass\_to\_teammate\_1, ..., pass\_to\_teammate\_11, Dribble\}$$

The features of state includes

- The absulute position the ball in current cycle and immediately previous cycle.
- The relative position of all players with respect to the ball in current cycle and immediately previous cycle.

The positions of ball are presented in Cartesian coordinates and all relative positions are presented in Polar coordinates. Figure 1 illustrates an instance at the moment of an opponent player possessing the ball in a soccer game.

We extract training and testing data from 99 games played between our agents and the champion of RoboCup 2004. We conduct these experiments to enable our agents to learn from the experience of playing with an opponent team. The statistics of these experiments is shown in Table 3 and Figure 2.

In this experiment, we also use the parameters tuning scheme applied in the experiment conducted on UCI datasets above. The experimental results are illustrated in Figure 3. In all four coding matrixes, our method outperforms $SVM^{light}$.

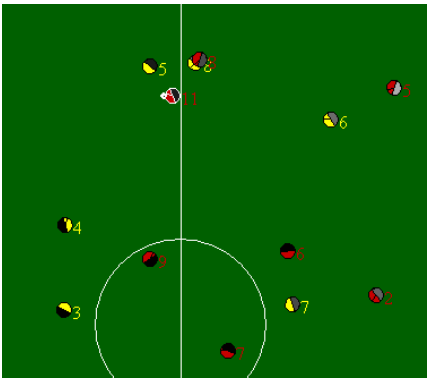


**Fig. 1.** Task of Robot Pass. Player 11 of opponent team (shown in red color) is possessing the ball, the task of RobotPass is to determine the next action of the player possessing ball. The potential action of opponent player 11 is dribbling or pass the ball to it's teammate 9 in the current situation.

**Table 3.** Statistics on RobotPass

| | #games | #instance | #pass | #dribbling |
|---|---|---|---|---|
| Train | 88 | 91109 | 16689 | 74420 |
| Test | 11 | 11440 | 2058 | 9382 |

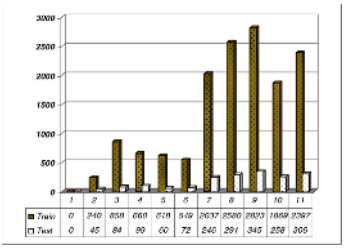


**Fig. 2.** The receive-passing frequency of each opponent in both training and testing data

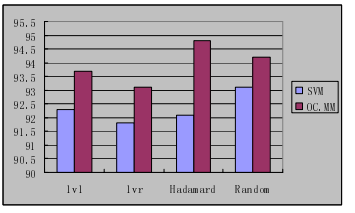


**Fig. 3.** Classification accuracy of SVM$^{light}$ (SVM) and our method (OC.MM)

## 5 Conclusions and Future Works

In this paper we present a novel version of ECOC framework which significantly boosts the performance of multi-class classification. We give a criteria of ECOC decoding by defining a global loss based on the empirical loss and regularization over the Extended Pair Samples. Empirical results on both UCI datasets and the task of opponent action prediction in RoboCup domain show the utility of our algorithm. We also notice that the performance improvement is more significant on the datasets which have more classes. This might be due to the limitation of conventional ECOC framework on complex data while it is overcome in our approach.

In spite of the presented effective training scheme of OC.MM, a large scale quadratic programming problem is still time-consuming. Although the training can be conducted off-line, the efficiency of optimization remains to be further improved in order to make our algorithm more practical in very large datasets. Another direction of future work is to conduct further statistical analysis on the OC.MM algorithm. In this novel ECOC framework, the problem of codewords selection remains open. But the introduction of decoding margin provides a potential direction of further statistical analysis such as upper bound of generalization using statistical learning theorems.

## Acknowledgement

This work is supported by the National Science Foundation of China (NSFC) No. 60305010 and No. 60421002.

## References

[1] Aha, D. W. (1997). Cloud classification using error-correcting output codes. *Artificial Intelligence Applications: Natural Science, Agriculture, and Environmental Science*, *11*, 13–28.

[2] Allwein, E., Schapire, R., & Singer, Y. (2000). Reducing multiclass to binary: A unifying approach for margin classifiers. *achine Learning: Proceedings of the Seventeenth International Conference. Artificial Intelligence Research, 2*, 263-286.

[3] Berger, A. (1999). Error-correcting output coding for text classification. In *IJCAI'99: Workshop on Machine Learning for Information Filtering*, In Berlin: Springer Verlag.

[4] Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and regression trees. Belmont,CA: Wadsworth & Brooks.

[5] Crammer, K.,Singer, Y. (2002). On the Learnability and Design of Output Codes for Multiclass Problems. *Machine Learning* 47(2-3):201-233.

[6] Crammer, K. & Singer. Y. (2001). On the algorithmic implementation of multiclass kernel-based machines. *Journal of Machine Learning Research, 2(Dec)*:265–292.

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

[8] Dietterich, T., & Kong, E. B. (1995). –Machine learning bias, statistical bias, and statistical variance of decision tree algorithms. Technical report, Oregon State University. Available via the WWW at http://www.cs.orst.edu:80/ tgd/cv/tr.html.

[9] Hastie, T. & Tibshirani, R. (1998). Classification by pairwise coupling. In *Advances in Neural Information Processing Systems*, volume 10. MIT Press.

[10] Hastie, T., Tibshirani, R., & Friedman, J. (2001). The Elements of Statistical Learning: data mining, inference and prediction. Springer-Verlag.

[11] Hsu, C.-W. & Lin, C.-J. (2002). A comparison of methods for multi-class support vector machines , *IEEE Transactions on Neural Networks, 13,* 415-425.

[12] Joachims, T. (2002). Optimizing Search Engines Using Clickthrough Data. *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*, ACM.

[13] Kuhlmann, G., & Stone, P. (2004). Progress in learning 3 vs. 2 keepaway. In Polani, D.; Browning, B.; Bonarini, A.; and Yoshida, K., eds., RoboCup-2003: Robot Soccer World Cup VII.

[14] Passerini, A., Pontil, M., & Frasconi, P.(2004). New results on error correcting output codes of kernel machines. *IEEE Transactions on Neural Networks, 15(1)*:45-54.

[15] Passerini, A.,Pontil, M., & Frasconi, F. (2002). From Margins to Probabilities in Multiclass Learning Problems. *ECAI:* 400-404.

[16] Quinlan, J. R. (1993). C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann.

[17] Rumelhart, D. E., Hinton, G. E.,&Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart, & J. L. McClelland (Eds.), Parallel distributed processing-explorations in the microstructure of cognition (ch. 8, pp. 318-362). Cambridge, MA: MIT Press.

[18] Schapire, R. E. (1997). Using output codes to boost multiclass learning problems. In Machine Learning. *Proceedings of the Fourteenth International Conference* (pp. 313-321).