Proceedings of the 2007 IEEE Symposium on
Foundations of Computational Intelligence (FOCI 2007)

Multiclass classification as a decoding problem

Takashi Takenouchi and Shin Ishii
Graduate School of Information Science Nara Institute of Science and Technology
8916-5, Takayama, Ikoma, Nara, Japan, 630-0192
Phone: +81-743-72-5378, FAX: +81-743-72-5989, E-mail: {ttakashi,ishii}@is.naist.jp

Abstract

In this article, we present a new method of multiclass classification by combining multiple binary classifiers in the context of information transmission theory. In the framework of the error correcting output coding (ECOC), a misclassification of each binary classifier is formulated as a bit inversion with a probabilistic model. While the conventional hamming decoding assumes the binary symmetric channel in a information transmission, the symmetric assumption is especially problematic in multiclass classification problems: for example, 1 vs R approach typically makes an asymmetric situation even if all classes contain the same number of examples. The asymmetry property corresponds to two kinds of error rate of the binary classification problem; the false positive error and the false negative error. We propose a probabilistic model which assumes an asymmetric channel having 3 inputs and 2 outputs.

By the maximum likelihood estimation with the proposed probabilistic model, we can identify properties of the noisy channel according to performances of applied binary classifiers. A multiclass label and a class membership probability for an input are easily estimated by the model. Experimental studies using a synthetic dataset and datasets from UCI repository are performed and results show that the proposed method is superior to the Hamming decoding and comparative to other multiclass classification methods such as multiclass Support vector machine.

1 Introduction

There are many methods to reduce a multiclass problem to multiple binary classification problems. Dietterich and Bakiri presented a general framework called error-correcting output coding (ECOC), in which a multiclass problem is decomposed into a number of binary classification problems [1]. Such decomposition is represented as a $p \times G$ code matrix W, where G and p are the number of classes and that of binary classification problems, respectively. Each class is represented as a p-dimensional vector called a code word. If $W_{jk} = 1 (= -1)$, where W_{jk} is the jk component of W, then the example feature vectors belonging to class k are regarded as positive (negative) examples for the j-th binary classification problem. The simplest method of ECOC, called the Hamming decoding, obtains the class prediction of a new feature vector according to Hamming distance between the estimated code of the feature vector and the code word of each class, but ignores the reliability of each predictor. Although the original ECOC did not allow the code matrix to contain zero components, meaning all examples are used in every binary classification problem, Allwein et al. extended the framework of ECOC such to allow the coding matrix to have 0 components and analyzed its theoretical aspects [2]. If $W_{jk} = 0$, examples belonging to the k-th class are not used in training the j-th classifier. They also presented the loss-based decoding which includes the Hamming decoding as a special case where the loss is associated with the Hamming distance.

Decoding in these methods primarily tries to assign a single class label to each example, but they do not output class membership probability estimate for an example. Such a probability estimate is important, because its representation of classification confidence is useful not only for considering various noises like mislabeling but also designing code words (i.e., coding). Hastie and Tibshirani presented a different approach to the decoding problem, which integrates multiple binary classifiers to obtain class probability membership estimates for a multiclass problem, given the code matrix W and a binary classification algorithm which outputs probability estimates [3]. They considered all-pairs (1 vs. 1) coding, where a classifier is trained for every pair of classes by ignoring the examples that do not belong to the class pair. Zadorozny extended this coding framework to applicable to arbitrary code matrices [4]. These two methods were based on the Bradley-Terry model (See [5]).

Although these Bradley-Terry model-based approaches can thus treat class membership probabilities, general estimation of probabilities is still problematic, because there are a little theoretical support for convergence of assignment of class probability into a unique solution except in the case of all-pairs (1 vs 1) [6]. In addition, while the Bradley-Terry model-based approaches requires probabilistic outputs from binary classifiers, most of the popular classifiers like SVM return binary class labels and then, another estimation process to get probabilistic outputs is necessary. . In this article, we propose an alternative but novel approach to the probabilistic decoding, which is based on a probabilistic model of information transmission.

1-4244-0703-6/07/$20.00 ©2007 IEEE

In section 2, basic settings and a probabilistic model of noisy channel are formulated. In section 3, we discuss the relationship between the proposed model and the conventional Hamming decoding. In section 4. performance of the proposed method is examined by comparing other multiclass classification methods using a synthetic dataset and UCI repository datasets.

2 Setting and method

In this section, we propose a new approach to multiclass classification problems. In subsection 2.1, a model of noisy channel in multiclass classification is mathematically formulated. We present a decoding method of class label disturbed by the noisy channel in subsection 2.2.

2.1 Information transmission through noisy channel and multiclass classification problem

In this study, we define a multiclass classification problem as the optimization of the decoder of information bits. Let $\boldsymbol{x} \in \mathrm{R}^p$ be a feature vector and $y \in \{1, \cdots, G\}$ be a label of the feature vector $\boldsymbol{x}$. The label y is represented as a G-dimensional vector $\boldsymbol{s}$ whose j-th component s_j is $\mathrm{I}(j = y)$, where $\mathrm{I}(\cdot)$ is an indicator function:

$$\mathrm{I}(R) = \begin{cases} 1 & \text{if } R \text{ is true} \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where R is an arbitrary conditional expression. We assume that a dataset $(X, Y) \equiv \{\boldsymbol{x}^i, y^i\}_{i=1}^n$ is given and let $\boldsymbol{s}^i$ be the representation of y^i and $S = \left(\boldsymbol{s}^1, \cdots, \boldsymbol{s}^n\right)$ the set of $\boldsymbol{s}^i$.

Now we consider the decomposition of a multiclass classification problem into multiple binary classification problems. Let $W \in \{1, 0, -1\}^{p \times G}$ be a coding matrix. The maximum number of binary classification problems p, is $(3^G - 2^{G+1} + 1)/2$, which exponentially grows as G becomes large, as shown in Table 1. Then, the coding matrix W is often restricted so that a subset in the full coding matrix W^* is used: in the 1 vs R method, each code word contains only one 1 (e.g. $(-1, 1, -1, \cdots, -1)$), or in all pairs (1 vs 1), each code word contains only one 1 and one -1 so that all other components are 0 (e.g. $(1, 0, -1, 0, \cdots, 0)$).

Table 1: The maximum number of binary classification problems for G-class problems.

G	2	3	4	5	6	$\cdots$	G
max p	1	6	25	90	301	$\cdots$	$\frac{3^G - 2^{G+1}+1}{2}$

For example, if $G = 3$, the full coding matrix W^* is given by

$$W^* = \begin{pmatrix} 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{pmatrix}. \tag{2}$$

Let $\boldsymbol{z} = (z_1, \cdots, z_p)'$ be a random variable vector, whose instance is given by a 'coded label' $\boldsymbol{z}^i = Ws^i$, which is usually a redundant representation of the label y_i. Let z_j^i be the j-th component of the code label $\boldsymbol{z}^i$ and $Z = WS$ be the set of coded labels. Note that the matrix Z has special degeneracy, because its column vectors have only G patterns.

Using the notations above, the multiclass classification problem is defined as follows: given a code matrix W, binary classifiers are trained based on the dataset (X, Z). A combination of the input set X and each row of the coded label matrix Z provides a dataset to one binary classification problem. Let $Z_{j\cdot} \equiv (z_j^1, \cdots, z_j^n)$ be the j-th row of Z. The $Z_{j\cdot}$ is the label set of the input set X.

To the j-th binary classification problem $(X, Z_{j\cdot})$, we can apply any binary classification algorithm such as SVM and AdaBoost. Training of the j-th binary classifier is executed by examples whose label z_j^i is either of 1 or -1 in the code word, but examples labeled as 0 are not used. After the training phase, examples labeled as 0 are also classified into 1 or -1 by the trained binary classifier. We describe these classified labels as $\tilde{Z}_{j\cdot} = (\tilde{z}_j^1, \cdots, \tilde{z}_j^n)$. By applying this procedure for all rows in Z, we obtain another matrix $\tilde{Z}$ whose j-th row is $\tilde{Z}_{j\cdot}$.

This process can be interpreted in the context of information transmission theory as follows (see figure 1). A component z_j of the coded label is transmitted into $\tilde{z}_j$ through a noisy channel. We assume that the noisy channel is memoryless and is not necessarily binary symmetric. Additionally, the distribution of noise associated with z_j is assumed to be different from the others. Here the memory-less assumption implies dependence of coded labels. Each component z_j^i of $\boldsymbol{z}^i$ is added a noise of bit inversion if z_j^i is 1 or -1, or transformed into 1 or -1 if z_j^i is 0, by the noisy channel. Asymmetric assumption corresponds to two kinds of error rate of the binary classification problem; the false positive error and the false negative error. Those errors are not generally equal, and this asymmetric property is prominent especially in multiclass classification problems: for example, 1 vs R approach typically makes an asymmetric situation even if all classes contain the same number of examples. Considering those situations leads to the assumption that each component z_j of the coded label is associated with a specific binary asymmetric channel.

The transmitted coded label $\tilde{z}_j$ is composed of 1 and -1. The property of the noisy channel is schematically shown in figure 2. The inversion probability of z_j from 1 to -1 is denoted as ε_{1j} and that from -1 to 1 is as ε_{2j}. The code 0 is

transformed to 1 with probability f_j. Those quantities represent the property of the noisy channel for z_j, and can be estimated with a probabilistic model according to the performance of applied binary classifiers.

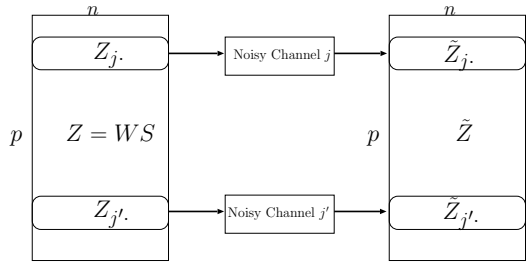

Figure 1: Transmission of the coded label matrix Z.

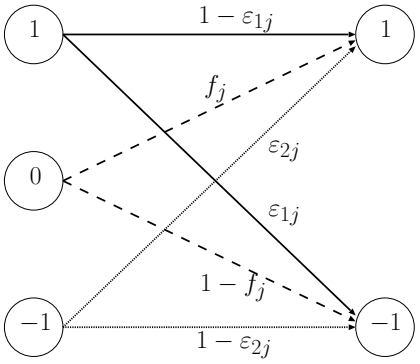

Figure 2: Property of the noisy channel for a j-th component z_j of the coded label.

Now we define a probabilistic model of information transmission of the j-th coded label z_j. Let $\tilde{z}_j$ be the transmitted z_j through the noisy channel, assumed to be a realization value of the random variable. The transmission of each component z_j of the coded label is modeled as a stochastic process:

$$p(\tilde{z}_j|z_j;\beta_j,\gamma_j,\delta_j) = \exp\left((\beta_j z_j+\gamma_j)\tilde{z}_j-\psi_1(z_j,\beta_j,\gamma_j)\right)^{z_j^2} \exp\left(\delta_j\tilde{z}_j-\psi_2(\delta_j)\right)^{1-z_j^2},$$

where β_j is a coefficient which is associated with the noise level of the channel, γ_j represents the degree of asymmetry of the channel and δ_j represents the rate of transformation when $z_j = 0$. The characteristic parameters of the noisy channel, ε_{1j}, ε_{2j} and f_j, are determined by β_j, γ_j and δ_j. $\psi_1(z_j,\beta_j,\gamma_j)$ and $\psi_2(\delta_j)$ are normalization terms defined by

$$\psi_1(z_j,\beta_j,\gamma_j) = \log\left(\sum_{z'\in\{1,-1\}}\exp(z'(\beta_j z_j+\gamma_j))\right),$$
$$\psi_2(\delta_j) = \log\left(\sum_{z'\in\{1,-1\}}\exp(z'\delta_j)\right). \quad (3)$$

Since we assume that each component of the coded label is transmitted independently, the transmission of the full coded label $\boldsymbol{z}$ is simply modeled as

$$p(\tilde{\boldsymbol{z}}|\boldsymbol{z};\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta}) = \prod_{j=1}^{p} p(\tilde{z}_j|z_j;\beta_j,\gamma_j,\delta_j) \quad (4)$$

$$= \exp\left[\tilde{\boldsymbol{z}}'\zeta(\boldsymbol{\beta}\boldsymbol{z}+\boldsymbol{\gamma}) + \tilde{\boldsymbol{z}}'(I-\zeta)\boldsymbol{\delta} - \sum_{j=1}^{p} z_j^2\psi_1(z_j,\beta_j,\gamma_j) - \sum_{j=1}^{p}(1-z_j^2)\psi_2(\delta_j)\right], \quad (5)$$

where $\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta},\boldsymbol{\zeta}$ are defined by

$$\boldsymbol{\beta} = \begin{pmatrix}\beta_1 & 0 & \cdots & 0\\ 0 & \beta_2 & \cdots & 0\\ \vdots & \vdots & & \vdots\\ 0 & 0 & \cdots & \beta_p\end{pmatrix}, \boldsymbol{\gamma} = \begin{pmatrix}\gamma_1\\ \gamma_2\\ \vdots\\ \gamma_p\end{pmatrix},$$
$$\boldsymbol{\delta} = \begin{pmatrix}\delta_1\\ \delta_2\\ \vdots\\ \delta_p\end{pmatrix}, \boldsymbol{\zeta} = \begin{pmatrix}z_1^2 & 0 & \cdots & 0\\ 0 & z_2^2 & \cdots & 0\\ \vdots & \vdots & & \vdots\\ 0 & 0 & \cdots & z_p^2\end{pmatrix}. \quad (6)$$

Note that we can extend the above model to assuming full or partly dependency between code z_j and z_k by off-diagonalization of the matrix $\boldsymbol{\beta}$.

If we employ all possible binary classifiers into the coding matrix W, the 'code length' p of the coded label experimentally increases as the class number G grows, which may be computationally intractable. Then, we need to introduce some constraints to the code dependency matrix $\boldsymbol{\beta}$ especially when considering the full or partly dependency of codes. In addition, the restriction on the code word z_j to select p rows from the full coding matrix W^* is effective not only for saving the computational cost but also for improvement in generalization. The latter problem is statistically the variable selection problem, or the 'coding problem'. These two problems are difficult and still pending in this study, though we can consider some heuristics such as regularization and backward (or forward) elimination of variables (and hence binary classifiers) based on cross-validation performance (e.g., [7]).

In this study, for simplicity, we assume independence of codes and the coding matrix W is given *a priori*. For a given coded label matrix Z and a transmitted code label matrix $\tilde{Z}$, the coefficients $\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta}$ can be estimated by maximizing the log-likelihood function:

$$L\left(\tilde{Z};\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta}\right) = \sum_{i=1}^{n}\log p(\tilde{\boldsymbol{z}}^i|\boldsymbol{z}^i). \quad (7)$$

By differentiating (7), we have

$$0 = \frac{\partial L}{\partial\beta_j} = \sum_{i=1}^{n}(z_j^i)^2\left(z_j^i\tilde{z}_j^i - z_j^i\frac{\exp(\beta_j z_j^i+\gamma_j)-\exp(-\beta_j z_j^i-\gamma_j)}{\exp(\beta_j z_j^i+\gamma_j)+\exp(-\beta_j z_j^i-\gamma_j)}\right),$$

$$0 = \frac{\partial L}{\partial \gamma_j} = \sum_{i=1}^{n} (z_j^i)^2 \left(\tilde{z}_j^i - \frac{\exp(\beta_j z_j^i + \gamma_j) - \exp(-\beta_j z_j^i - \gamma_j)}{\exp(\beta_j z_j^i + \gamma_j) + \exp(-\beta_j z_j^i - \gamma_j)} \right),$$

$$0 = \frac{\partial L}{\partial \delta_j} = \sum_{i=1}^{n} (1 - (z_j^i)^2) \left(\tilde{z}_j^i - \frac{\exp(\delta_j) - \exp(-\delta_j)}{\exp(\delta_j) + \exp(-\delta_j)} \right).$$

The solution of those equations is given as

$$\hat{\beta}_j = \frac{1}{4} \log \frac{(1-\tilde{\varepsilon}_{1j})(1-\tilde{\varepsilon}_{2j})}{\tilde{\varepsilon}_{1j}\tilde{\varepsilon}_{2j}}, \hat{\gamma}_j = \frac{1}{4} \log \frac{(1-\tilde{\varepsilon}_{1j})\tilde{\varepsilon}_{2j}}{\tilde{\varepsilon}_{1j}(1-\tilde{\varepsilon}_{2j})},$$

$$\hat{\delta}_j = \frac{1}{2} \log \frac{\tilde{f}_j}{1-\tilde{f}_j}, \tag{8}$$

where

$$\tilde{\varepsilon}_{1j} = \frac{\sum_{i=1}^{n} \mathrm{I}(z_j^i = 1, \tilde{z}_i^i = -1)}{\sum_{i=1}^{n} \mathrm{I}(z_j^i = 1)},$$

$$\tilde{\varepsilon}_{2j} = \frac{\sum_{i=1}^{n} \mathrm{I}(z_j^i = -1, \tilde{z}_i^i = 1)}{\sum_{i=1}^{n} \mathrm{I}(z_j^i = -1)}, \tilde{f}_j = \frac{\sum_{i=1}^{n} \mathrm{I}(z_j^i = 0, \tilde{z}_j^i = 1)}{\sum_{i=1}^{n} \mathrm{I}(z_j^i = 0)}.$$

The coefficient $\hat{\beta}_j$ becomes large if the error levels, $\tilde{\varepsilon}_{1j}$ and $\tilde{\varepsilon}_{2j}$, of the noisy channel are both low. The absolute value of the coefficient $\hat{\gamma}_j$ becomes large when the noisy channel is highly asymmetric, presumably corresponding to unbalanced problems, and the coefficient δ_j represents the asymmetry level of the noisy channel for the zero code ($z_j = 0$).

2.2 Decoding of label

After identifying the property of the noisy channel by estimating its characteristic parameters, we can decode the original coded label $\boldsymbol{z}$ from the transmitted label $\tilde{\boldsymbol{z}}$. The decoding is executed by calculating the posterior probability $p(\boldsymbol{z}|\tilde{\boldsymbol{z}})$ of the coded label $\boldsymbol{z}$, and searching for a coded label that maximizes the posterior probability. Under the usual situation, there could be 2^p candidates to search and therefore, Maximum a posteriori (MAP) decoding is difficult. Then Maximization of the posterior marginals (MPM) approach is applied with the belief propagation method [8]. In our multiclass classification problem, on the other hand, we can apply MAP decoding because the number of candidates to be searched is only G, whereas the dimension of code p can be exponentially large. The prior distribution $p(\boldsymbol{z})$ of the coded label $\boldsymbol{z}$ can be estimated as

$$p(\boldsymbol{z}) = \begin{cases} \frac{1}{n} \sum_{i=1}^{n} \mathrm{I}(y^i = j) & \text{if } \boldsymbol{z} = W_{\cdot j} \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

After the model of noisy channel is obtained, we can predict the label of a new feature $\boldsymbol{x}^{new}$ as follows:

1. The noisy coded label $\tilde{\boldsymbol{z}}^{new}$ for $\boldsymbol{x}^{new}$ is obtained from the set of trained binary classifiers.
2. Calculate the posterior probability as
$$p(\boldsymbol{z}|\tilde{\boldsymbol{z}}^{new}) = \frac{p(\tilde{\boldsymbol{z}}^{new}|\boldsymbol{z}; \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\gamma}}, \hat{\boldsymbol{\delta}}) p(\boldsymbol{z})}{p(\tilde{\boldsymbol{z}}^{new}; \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\gamma}}, \hat{\boldsymbol{\delta}})}. \tag{10}$$
3. Reconstruct the 'noise-less' code by $\mathrm{argmax}_{\boldsymbol{z}}\, p(\boldsymbol{z}|\tilde{\boldsymbol{z}}^{new}; \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\gamma}}, \hat{\boldsymbol{\delta}})$.

Note that maximization operation is performed by comparing G codes with the consideration of the prior $p(\boldsymbol{z})$ and this decoding process is computationally easy. Note also that the posterior $p(\boldsymbol{z}|\tilde{\boldsymbol{z}})$ provides the class membership probability estimates for the feature vector $\boldsymbol{x}$, which could be used for managing misclassification risk.

3 Relationship between the proposed method and the hamming decoder

If we set $\gamma_j = 0, \delta_j = 0$ for all j and $p(\boldsymbol{z})$ is uniform over possible G classes, then the proposed method reduces to the weighted Hamming decoder, which returns the label that minimizes the weighted Hamming distance between the transmitted code and each code word in the coding matrix W. The posterior probability of $\boldsymbol{z}$ is

$$p(\boldsymbol{z}|\tilde{\boldsymbol{z}}) \propto \exp\left(\sum_{j=1}^{p} \beta_j \tilde{z}_j z_j \right). \tag{11}$$

Note that the normalization term does not depend on $\boldsymbol{z}$ or $\tilde{\boldsymbol{z}}$ in this setting. The maximization of this posterior is equivalent to the minimization of the weighted Hamming distance:

$$\sum_{j=1}^{p} \beta_j \frac{1 - z_j \tilde{z}_j}{2}. \tag{12}$$

4 Simulation study

In this section, we examine the performance of the proposed method by comparing with the Hamming decoding and two implementations of multiclass support vector machine (see [9]) with a polynomial kernel and an rbf kernel, using a synthetic dataset and UCI repository datasets. In our method, we used SVM as individual binary classifiers and the full coding matrix W^* as the coding matrix W.

4.1 Synthetic dataset

The proposed method was first applied to the following synthetic dataset. The number of classes, G, was four and $\boldsymbol{x} = (x_1, x_2)'$ was uniformly distributed on $[-1, 1]^2$. The

label y was *a priori* distributed as $p(y) = 1/4$, so that the true label y was given by

$$y = \begin{cases} 1 & -1 \le x_2 < -0.5 \\ 2 & -0.5 \le x_2 < 0 \\ 3 & 0 \le x_2 < 0.5 \\ 4 & 0.5 \le x_2 \le 1, \end{cases} \tag{13}$$

but was assumed to be observed after randomly shuffled with probability 0.3. A typical dataset containing 200 examples is shown in figure 3. When applied to this dataset, the proposed

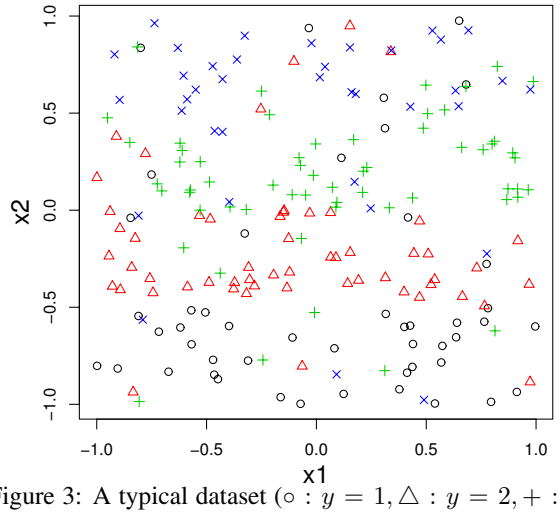

Figure 3: A typical dataset ($\circ : y = 1, \triangle : y = 2, + : y = 3, \times : y = 4$).

method could easily estimate the class membership probability for each example (and each point on the domain). The probability along $x_1 = 0$, $p(y|x_1 = 0, x_2)$, estimated by 200 examples is shown in figure 4. When the probability of label shuffling was lower, the class boundaries were estimated more sharply (data not shown). We repeatedly generated 100 pairs of a training dataset and a test dataset each containing 200 and 5000 examples, respectively, and examined the average performance. For comparison, we applied C-svm, spoc-svm, which are implementation variants of multiclass SVM. Means, and 5% and 95% percentiles among 100 trials of training error and test error are shown in figure 5 and 6. We can observe that the proposed method in average outperformed C-svm and spoc-svm with respect to the test error regardless of the kernel function employed. About the training error, the proposed method was likely to attain reasonable results. Moreover, error variance of the proposed method was smaller than those of the other methods, suggesting the stability of the classification performance.

4.2 UCI repository

We next applied the proposed method to some datasets (Table 2) registered in the UCI repository. Each of the three datasets from the top are originally separated into a training set and a

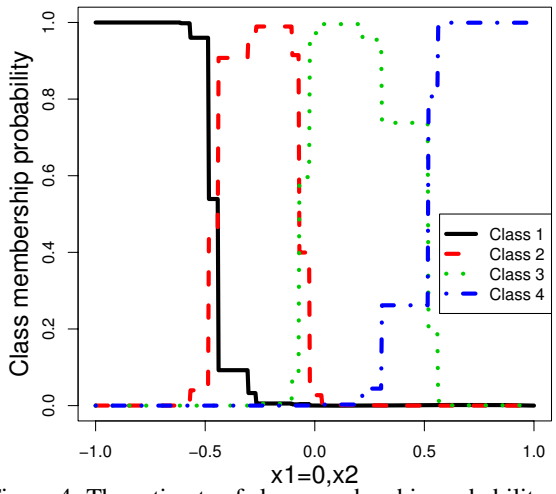

Figure 4: The estimate of class membership probability along $x_1 = 0$.

Table 2: Information of UCI datasets.

Dataset	#Examples	#Attributes	#lCasses
Ann-thyroid	7200	21	3
Satimage	6435	36	6
Segmentation	2310	19	7
Iris	150	4	3
Wine	178	3	3
Glass	214	9	6

test set. For the other datasets, we applied 5-fold cross validation to estimate the generalization performance; we repeated the 5-fold cross validation 100 times and observed the average performance.

In Table 3, averaged generalization performance is shown and a method with the best performance for each dataset is boldfaced. For the dataset to which we applied 5-fold cross-validation, the average of 100 trials and its standard deviation are shown. While a superior method and a good kernel function depend on datasets, all methods have attained reasonably low error rates. We can see, however, that the proposed method is comparative to or outperforms the multiclass SVM implementations.

5 Conclusion

We proposed an integration method of binary classifiers to construct a multiclass classifier based on a probabilistic model of information transmission. The probabilistic formulation of a noisy channel was presented, which allowed the original multicalss class label to be decoded from the transmitted and hence noisy code output. The method includes the weighted Hamming decoding as a special case and the decoding of the proposed method was easy to implement. Moreover, the method was likely to outperform the existing

Table 3: Generalization error rate for UCI repository datasets

Method	Spoc-svm:Rbf(Poly.)	C-svm:Rbf(Poly.)	Ham.:Rbf(Poly.)	Prop.1:Rbf(Poly.)	Prop.2:Rbf(Poly.)
Thy.	$0.0502(0.0414)$	$0.0534(0.0338)$	$0.0524(0.0368)$	$\mathbf{0.0324}(0.0333)$	$0.0327(0.0335)$
Sat.	$\mathbf{0.1175}(0.1640)$	$0.1185(0.1405)$	$0.1250(0.1643)$	$0.1210(0.1455)$	$0.1180(0.1465)$
Seg.	$0.1419(0.0738)$	$0.1762(\mathbf{0.0581})$	$0.1647(0.0855)$	$0.1557(0.0762)$	$0.1710(0.1381)$
Iris	0.0419 ± 0.0083	0.0404 ± 0.0081	0.0405 ± 0.0076	0.0395 ± 0.0077	0.0405 ± 0.0074
	(0.0363 ± 0.0075)	(0.0371 ± 0.0075)	(0.0365 ± 0.0080)	0.0375 ± 0.0092	$(\mathbf{0.0359 \pm 0.0079})$
Wine	0.0171 ± 0.0084	0.0133 ± 0.0041	$\mathbf{0.0127 \pm 0.0035}$	0.0129 ± 0.0039	0.0140 ± 0.0050
	(0.0325 ± 0.0084)	(0.0383 ± 0.0085)	(0.0329 ± 0.0082)	(0.0331 ± 0.0086)	(0.0328 ± 0.0104)
Glass	0.1516 ± 0.0134	0.1646 ± 0.0131	0.1536 ± 0.0128	0.1222 ± 0.0131	0.0945 ± 0.0142
	(0.0762 ± 0.0128)	(0.0661 ± 0.0113)	(0.0873 ± 0.0132)	(0.0463 ± 0.0086)	$(\mathbf{0.0450 \pm 0.0096})$

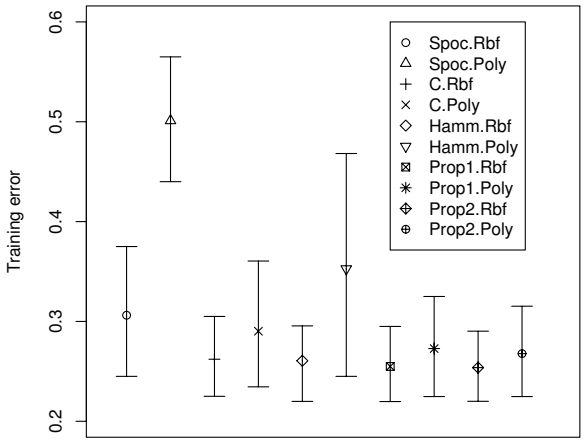

Figure 5: Average performance, and 5% and 95% percentiles over the training error of 100 trials for each method. Spoc-svm, C-svm and the proposed method with an rbf kernel and a polynomial kernel were applied.

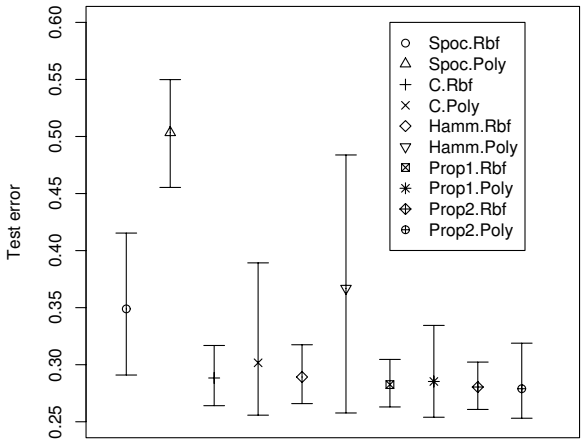

Figure 6: Average performance, and 5% and 95% percentiles over the test error of 100 trials for each method. Spoc-svm, C-svm and the proposed method with an rbf kernel and a polynomial kernel were applied.

multiclass classification methods such as Hamming decoding, spoc-svm and C-svm. An extension to a model considering dependency between binary classifiers, regularization for sparseness condition and introduction of the Bayesian framework, which correspond to the optimization in coding, are future work.

References

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

[2] E. L. Allwein, R. E. Schapire, and Y. Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113–141, 2001.

[3] T. Hastie and R. Tibshirani. Classification by pairwise coupling. *Annals of Statistics*, 26:451–471, 1998.

[4] B. Zadrozny. Reducing multiclass to binary by coupling probability estimates. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14, Cambridge, MA, 2001. The MIT Press.

[5] R. Bradley and M. Terry. Rank analysis of incomplete block designs, i:the method of paired comparisons. *Biometrics*, 39:324–345, 1952.

[6] T.-K. Huang, R.-C. Weng, and C.-J. Lin. Generalized bradley-terry models and multi-class probability estimatesgood error-correcting codes based on very sparse matrices. *Journal of Machine Learning Research*, 7:85–115, 2006.

[7] T. Hastie, R. Tibishirani, and J. Friedman. *The elements of statistical learning*. Springer, New York, 2001.

[8] D. J. C. MacKay. Good error-correcting codes based on very sparse matrices. *IEEE Transactions on Information Theory*, 45:399–431, 1999.

[9] K. Crammer and K. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001.