Research Memorandum No. 1010 September 28, 2006

Learning Binary Classifiers for Multi-Class Problem

Shiro Ikeda

The Institute of
Statistical Mathematics

4-6-7 Minami-Azabu, Minato-ku,
Tokyo, 106-8569, Japan

Learning Binary Classifiers for Multi-Class Problem

Shiro Ikeda
The Institute of Statistical Mathematics
Tokyo, Japan 106-8569
shiro@ism.ac.jp

28th September, 2006

Abstract

One important idea for the multi-class classification problem is to combine binary classifiers (base classifiers), which is summarized as error correcting output codes (ECOC), and the generalized Bradley-Terry (GBT) model gives a method to estimate the multi-class probability. In this memo, we review the multi-class problem with the GBT model and discuss two issues. First, a new estimation algorithm of the GBT model associated with α–divergence is proposed. Secondly, the maximum likelihood estimation (MLE) algorithm of each base classifier based on the combined multi-class probability is derived.

1 Introduction

The idea of ECOC [5] is to combine the outputs of base classifiers for multi-class classification problems, where each base classifier is prepared to separate two disjoint subsets of classes. Many studies have appeared on this subject [1, 9, 10]. One of the reasons of the interests is the developments of SVM and AdaBoost. These classifiers work surprisingly well for binary classification problems but its extension to multi-class is not necessarily straightforward.

The basic approach of ECOC is to compute the Hamming distances between the binary outputs of base classifiers and each desired codewords. Although it is sufficient for classification, let us assume that the output of each base classifier is not binary but a probability prediction. Now our problem is to combine the predictions for multi-class probability estimation. One interesting proposal [6] was to use the Bradley-Terry (BT) model [3]. In the original proposal, 1-vs-1 classifiers were used as the base classifiers. Then, it has been generalized to wider class of base classifiers, and the BT model is generalized to the GBT (generalized Bradley-Terry) models [8].

In this article, we first review the GBT model for multi-class probability estimation and propose two extensions. One is concerning the estimation of the GBT model. For combining classifiers, it is not necessary to use the likelihood function, which gives the MLE for the original BT model. We propose a new estimation algorithm associated with α-divergence [2]. The algorithm is similar to the EM algorithm [4]. The other extension is the parameter estimation of each base classifier. In ECOC, each base classifier is prepared beforehand. But it is better to tune them to make the combined outputs better. We derive a MLE method for the base classifiers based on the combined multi-class probability. The algorithm can be applied to any differentiable model. We first consider the logistic regressions (LRs), and then extend it to the kernel logistic regressions (KLRs) [11].

2 Generalized Bradley-Terry Model and Multi-class Probability Estimation

2.1 Original Bradley-Terry Model

We briefly review the BT and GBT models which will be used to combine base classifiers' outputs. We start with the BT model [3]. Let us assume there are K players and each plays games against another. The BT model assumes the probability of player i to beat player j is given by

$$P(i \ \text{ beats } \ j) = \frac{p_i}{p_i + p_j}, \quad p_i, p_j > 0, \quad i \neq j, \quad i, j = 1, \cdots, K.$$

This is scale invariant, and we assume $\sum_j p_j = 1$. When we observe independent results of games

$$n_{ij} = \#(\text{games between } i \text{ and } j), \quad r_{ij} = \frac{\#(\ i \ \text{ beats } \ j)}{n_{ij}}, \quad r_{ij} + r_{ji} = 1,$$

the likelihood function is given as

$$l(\boldsymbol{p}) = \sum_{i<j} n_{ij} \Big(r_{ij} \log \frac{p_i}{p_i + p_j} + r_{ji} \log \frac{p_j}{p_i + p_j} \Big),$$

where $\boldsymbol{p} = (p_1, \cdots, p_K)$, and the MLE is the maximizer of $l(\boldsymbol{p})$.

2.2 Generalized Bradley-Terry Models

In the GBT model, we assume two teams, I_a^+ and I_a^-, play games. Teams are defined as

$$I_a^+, I_a^- \subset \{1, \cdots, K\}, \quad I_a^+, I_a^- \neq \emptyset, \quad I_a^+ \cap I_a^- = \emptyset, \quad I_a = I_a^+ \cup I_a^-.$$

The GBT model assumes the probability of I_a^+ to beat I_a^- is given by

$$P(I_a^+ \ \text{ beats } \ I_a^-) = \frac{\sum_{j \in I_a^+} p_j}{\sum_{j \in I_a} p_j}, \quad a = 1, \cdots, M,$$

where p_i's are the same as those in the original BT model. The observed results of games are

$$n_a = \#(\text{games between } I_a^+ \text{ and } I_a^-), \quad r_a = \frac{\#(I_a^+ \text{ beats } I_a^-)}{n_a}, \quad r'_a = \frac{\#(I_a^- \text{ beats } I_a^+)}{n_a},$$

where $r_a + r'_a = 1$. The likelihood function is the following $l(\boldsymbol{p})$, and the maximizer is the MLE.

$$l(\boldsymbol{p}) = \sum_{a=1}^{M} n_a \Big(r_a \log \frac{q_a^+}{q_a} + r'_a \log \frac{q_a^-}{q_a} \Big), \quad q_a^+ = \sum_{j \in I_a^+} p_j, \quad q_a^- = \sum_{j \in I_a^-} p_j, \quad q_a = q_a^+ + q_a^-.$$

2.3 Estimation Algorithm and Convergence

Huang et al.[8] discussed the MLE algorithm and its convergence. We show their results briefly.

Algorithm 1 *(Algorithm 2 in [8])*

1. *initialize* $\{p_i^0\}$ *and compute* $\{q_a^{0,+}\}$, $\{q_a^{0,-}\}$ *and* $\{q_a^0\}$.

2. *repeat the following by increasing* $t = 0, 1, \cdots$, *until convergence*

   (a) $i = (t \mod K) + 1$.

$$p_i^{t+1} = \frac{\sum_{a: i \in I_a^+} \frac{r_a}{q_a^{t,+}} + \sum_{a: i \in I_a^-} \frac{r'_a}{q_a^{t,-}}}{\sum_{a: i \in I_a} \frac{r_a + r'_a}{q_a^t}} p_i^t, \quad p_j^{t+1} = p_j^t, \quad \forall j \neq i$$

   (b) *normalize* $\boldsymbol{p}^{t+1}$ *and update* $\{q_a^{t+1,+}\}$, $\{q_a^{t+1,-}\}$ *and* $\{q_a^{t+1}\}$.

They have discussed the condition of the algorithm to increase the likelihood function (Theorem 1 in [8]), and make the following assumption

Assumption 1. *For any* $i, j \in \{1, \cdots, K\}$, $i \neq j$, *there are* $I_{a_0}, \cdots, I_{a_t}$ *such that either*

$$\begin{cases} 1. \quad r_{a_0} > 0, r_{a_1} > 0, \cdots, r_{a_t} > 0, \\ 2. \quad I_{a_0}^+ = \{i\}; I_{a_r}^+ \subset I_{a_{r-1}}, r = 1, \cdots, t; \\ \quad\; j \in I_{a_t}^- \end{cases} \text{ or } \begin{cases} 1. \quad r'_{a_0} > 0, r'_{a_1} > 0, \cdots, r'_{a_t} > 0, \\ 2. \quad I_{a_0}^- = \{i\}; I_{a_r}^- \subset I_{a_{r-1}}, r = 1, \cdots, t; \\ \quad\; j \in I_{a_t}^+. \end{cases}$$

And further they derived the following theorem (Theorem 4 in [8]).

Theorem 1. *Under the Assumption 1, if either* $|I_a^+| = |I_a^-|$, $\forall a$ *or* $|I_a| = K$, $\forall a$ *then the algorithm converges to a unique MLE.*

We omit the details which can be found in [8]. In section 3, we will propose a new algorithm and will use these results.

2.4 Multi-class Probability Estimation

Let $\mathcal{X} = \Re^n$ be the feature space, $\mathcal{Y} = \{1, \cdots, K\}$ be the labels, and $(\boldsymbol{X}, Y) \in (\mathcal{X}, \mathcal{Y})$ be the random variables. Our problem is to estimate the probability of $Y = y$ when $\boldsymbol{X} = \boldsymbol{x}$ is given, that is, $P(y|\boldsymbol{x})$. We assume data follows $P(y, \boldsymbol{x}) = P(y|\boldsymbol{x})P(\boldsymbol{x})$ and $P(y|\boldsymbol{x}) > 0$ if $P(\boldsymbol{x}) > 0$.

Let us consider the case we have M base classifiers, each of which predicts $P(y \in I_a^+ |\ y \in I_a, \boldsymbol{x})$. In the following, we assume $|I_a| = K$, that is, $I_a = \mathcal{Y}$. This is not common in ECOC, but we employ this from the following reason: Although our problem is to estimate the multi-class probability when $\boldsymbol{x} \sim P(\boldsymbol{x})$ is given, the training data distribution of a base classifier is usually $P(\boldsymbol{x}|y \in I_a)$. Thus, outputs of base classifiers would be biased for $\boldsymbol{x} \sim P(\boldsymbol{x})$. These biases might be canceled when we use many base classifiers, and this is what we expect in ECOC. However, it may not work when M is small. Instead of relying on the random effect, we assume $|I_a| = K$. Under this assumption $y \in I_a$ is always true, and each base classifier predicts $P(y \in I_a^+|\boldsymbol{x})$. The base classifiers are defined as

$$r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a) = r_a(y \in I_a^+ |\ \boldsymbol{x}; \boldsymbol{\theta}_a), \quad a = 1, \cdots, M, \quad \boldsymbol{\theta}_a \in \Theta, \quad \text{where} \quad 0 < r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a) < 1,$$

here $\boldsymbol{\theta}_a$ is the parameter and we assume $r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a)$ is differentiable. Note that $1 - r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a)$ is the prediction of $(1 - P(y \in I_a^+ |\ \boldsymbol{x})) = P(y \in I_a^- |\ \boldsymbol{x})$.

When a data $\boldsymbol{x}$ is given, $r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a)$, $a = 1, \cdots, M$, are computed. Then we combine them as $p_{y|\boldsymbol{x}} = p(y|\boldsymbol{x})$, $\sum_{y=1}^K p_{y|\boldsymbol{x}} = 1$, which is computed by maximizing

$$l(\boldsymbol{p}_{|\boldsymbol{x}}) = \sum_a^M \Big(r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a) \log q^+_{a|\boldsymbol{x}} + (1 - r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a)) \log (1 - q^+_{a|\boldsymbol{x}}) \Big), \quad q^+_{a|\boldsymbol{x}} = \sum_{y \in I_a^+} p_{y|\boldsymbol{x}},$$

where, $\boldsymbol{p}_{|\boldsymbol{x}} = (p_{1|\boldsymbol{x}}, \cdots, p_{K|\boldsymbol{x}})$. This can be reformulated as minimizing the following function

$$\begin{aligned} \mathcal{D}(\boldsymbol{p}_{|\boldsymbol{x}}) &= \sum_{a=1}^M \Big(r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a) \log r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a) + (1 - r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a)) \log (1 - r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a)) \Big) - l(\boldsymbol{p}_{|\boldsymbol{x}}) \\ &= \sum_{a=1}^M D_{KL,a}\big[r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a); q^+_{a|\boldsymbol{x}} \big], \end{aligned} \tag{1}$$

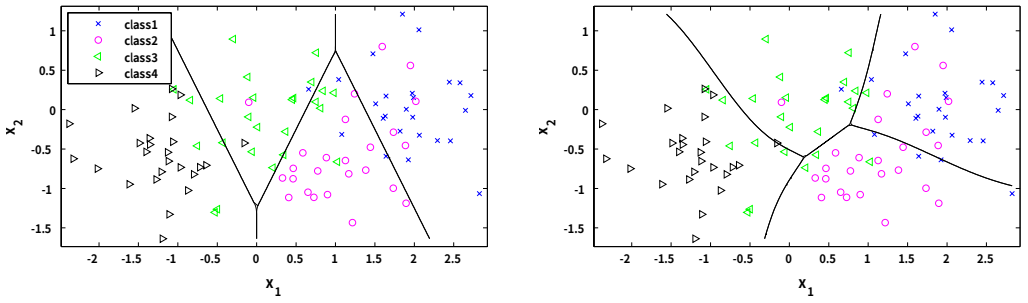
where $D_{KL,a}[r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a); q^+_{a|\boldsymbol{x}}]$ is the Kullback-Leibler (KL) divergence defined as

$$D_{KL,a}\big[r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a); q^+_{a|\boldsymbol{x}} \big] = r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a) \log \frac{r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a)}{q^+_{a|\boldsymbol{x}}} + (1 - r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a)) \log \frac{(1 - r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a))}{(1 - q^+_{a|\boldsymbol{x}})}.$$

Note $D_{KL,a}[r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a); q^+_{a|\boldsymbol{x}}] \geq 0$ and equality holds iff $r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a) = q^+_{a|\boldsymbol{x}}$. We choose the set of base classifiers to satisfy Assumption 1. Since $|I_a| = K$ holds, theorem 1 shows there is a unique $\boldsymbol{p}_{|\boldsymbol{x}}$ which minimizes eq.(1). Note that $\boldsymbol{p}_{|\boldsymbol{x}}$ is computed for each $\boldsymbol{x}$ separately.

Figure 1(a) is an example where $K = 4$. The data and the optimal decision boundary are shown. Each class of the data is drawn from a 2 dimensional normal distribution with the same variance-covariance

matrix, therefore optimal decision boundary is a collection of line segments. There are $(2^{(K-1)} - 1)$ possible binary classifiers, which is 7 in this case. We estimated those 7 classifiers with LRs shown in Fig. 2. The decision boundary based on the combined estimate is shown in Fig. 1(b). The boundaries of a binary LR and that of a multi-class LR (see for example [7]) are affine planes. But obviously Fig. 1(b) gives a different boundary. This is similar to the results in [6].



(a) (b)

Figure 1: (a) 4 classes data. The distribution of each class is a 2-dimensional normal distribution. The variance-covariance matrices are the same as 0.3 times identity matrix. Mean vectors are $(2, 0)$ for class 1, $(1, -0.5)$ for class 2, $(0, 0)$ for class 3, and $(-1, -0.5)$ for class 4. For training, 25 data are drawn from each class. The optimal decision boundary is computed from the true distribution. (b) Training data and the decision boundaries obtained by combining 7 classifiers given in Fig. 2.

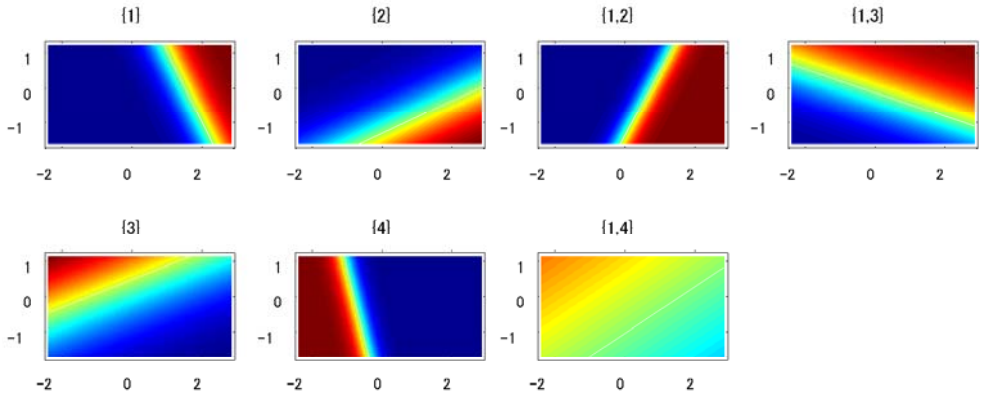


Figure 2: 7 base classifiers: 7 logistic regressions trained with the data in Fig. 1(a). I_a^+ is shown on the top. Colors show the output $r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a)$, where red is close to 1 and blue is close to 0.

3 Extensions

3.1 Modified Algorithm

The base classifiers are combined to $\boldsymbol{p}_{|\boldsymbol{x}}$ by minimizing eq.(1). We propose a slightly modified version of Algorithm 1. Let us start by defining the following set of conditional distributions.

$$\mathcal{S} = \Big\{ u(y|\boldsymbol{x}) \;\Big|\; \sum_{y \in \mathcal{Y}} u(y|\boldsymbol{x}) = 1, u(y|\boldsymbol{x}) > 0 \Big\}.$$

And let us consider the following subset $\mathcal{M}_a \subset \mathcal{S}$, $a = 1, \cdots, M$.

$$\mathcal{M}_a = \Big\{ u(y|\boldsymbol{x}) \;\Big|\; u(y|\boldsymbol{x}) \in \mathcal{S}, \sum_{y \in I_a^+} u(y|\boldsymbol{x}) = r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a), \Big\}.$$

Let the current estimate be $\boldsymbol{p}_{|\boldsymbol{x}}$, and let us consider the following problem,

$$\boldsymbol{u}^a_{|\boldsymbol{x}} = \underset{\boldsymbol{u}_{|\boldsymbol{x}} \in \mathcal{M}_a}{\operatorname{argmin}} D_{KL,y}[\boldsymbol{u}_{|\boldsymbol{x}}; \boldsymbol{p}_{|\boldsymbol{x}}], \tag{2}$$

where $D_{KL,y}[\boldsymbol{u}_{|\boldsymbol{x}}; \boldsymbol{p}_{|\boldsymbol{x}}] = \sum_{y=1}^K u_{y|\boldsymbol{x}} (\log u_{y|\boldsymbol{x}} - \log p_{y|\boldsymbol{x}})$. This is called e–projection from $\boldsymbol{p}_{|\boldsymbol{x}} \in \mathcal{S}$ to $\mathcal{M}_a$ in the information geometry [2]. The optimization in eq.(2) is equivalent to minimize $D_{KL,y}[\boldsymbol{u}_{|\boldsymbol{x}}; \boldsymbol{p}_{|\boldsymbol{x}}]$, subject to $\sum_{y \in I_a^+} u_{y|\boldsymbol{x}} = r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a)$. This minimization is simply solved by introducing Lagrange multipliers and the minimizer becomes

$$u^a_{y|\boldsymbol{x}} = \begin{cases} r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a) \dfrac{p_{y|\boldsymbol{x}}}{\sum_{y' \in I_a^+} p_{y'|\boldsymbol{x}}}, & y \in I_a^+ \\ (1 - r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a)) \dfrac{p_{y|\boldsymbol{x}}}{1 - \sum_{y' \in I_a^+} p_{y'|\boldsymbol{x}}}, & y \notin I_a^+ \end{cases} \tag{3}$$

Interestingly, $D_{KL,y}[\boldsymbol{u}^a_{|\boldsymbol{x}}; \boldsymbol{p}_{|\boldsymbol{x}}] = D_{KL,a}[r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a); q^+_{a|\boldsymbol{x}}]$ holds. Now we propose the following algorithm.

Algorithm 2

1. *initialize* $\boldsymbol{p}^0_{|\boldsymbol{x}}$
2. *repeat the following by increasing* $t = 0, 1, \cdots,$ *until convergence*
   (a) *compute* $\boldsymbol{u}^{a,t}_{|\boldsymbol{x}}$, $a = 1, \cdots, M$ *(practical computation is eq.*(3)*),*

$$\boldsymbol{u}^{a,t}_{|\boldsymbol{x}} = \underset{\boldsymbol{u}_{|\boldsymbol{x}} \in \mathcal{M}_a}{\operatorname{argmin}} D_{KL,y}[\boldsymbol{u}_{|\boldsymbol{x}}; \boldsymbol{p}^t_{|\boldsymbol{x}}].$$

   (b) *compute* $\boldsymbol{p}^{t+1}_{|\boldsymbol{x}}$ *which minimizes*

$$\boldsymbol{p}^{t+1}_{|\boldsymbol{x}} = \underset{p_{|\boldsymbol{x}} \in \mathcal{S}}{\operatorname{argmin}} \sum_{a=1}^M D_{KL,y}[\boldsymbol{u}^{a,t}_{|\boldsymbol{x}}; \boldsymbol{p}_{|\boldsymbol{x}}].$$

The minimization of step 2(b) is also solved by introducing Lagrange multipliers as follows

$$\boldsymbol{p}_{|\boldsymbol{x}}^{t+1} = \frac{1}{M}\sum_{a=1}^{M} \boldsymbol{u}_{|\boldsymbol{x}}^{a,t}, \tag{4}$$

which is called m–mixture. Now we see that the cost function in eq.(1) decreases at each step,

$$\begin{aligned}\mathcal{D}(\boldsymbol{p}_{|\boldsymbol{x}}^{t}) &= \sum_{a=1}^{M} D_{KL,a}[r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a); q_{a|\boldsymbol{x}}^{+,t}] = \sum_{a=1}^{M} D_{KL,y}[\boldsymbol{u}_{|\boldsymbol{x}}^{a,t}; \boldsymbol{p}_{|\boldsymbol{x}}^{t}] \geq \sum_{a=1}^{M} D_{KL,y}[\boldsymbol{u}_{|\boldsymbol{x}}^{a,t}; \boldsymbol{p}_{|\boldsymbol{x}}^{t+1}] \\ &\geq \sum_{a=1}^{M} D_{KL,y}[\boldsymbol{u}_{|\boldsymbol{x}}^{a,t+1}; \boldsymbol{p}_{|\boldsymbol{x}}^{t+1}] = \mathcal{D}(\boldsymbol{p}_{|\boldsymbol{x}}^{t+1}), \quad \text{where } \; q_{a|\boldsymbol{x}}^{+,t} = \sum_{y\in I_a^+} p_{y|\boldsymbol{x}}^{t}.\end{aligned}$$

The idea of the proof is similar to that of the EM algorithm [4]. Moreover if $\boldsymbol{p}_{|\boldsymbol{x}}^{t+1} \neq \boldsymbol{p}_{|\boldsymbol{x}}^{t}$, $\mathcal{D}(\boldsymbol{p}_{|\boldsymbol{x}})$ strictly decreases and since we assume there is a unique minimum, this algorithm always converges to the global minimum.

3.2 New Cost Function and Algorithm

The cost in eq.(1) is inspired by the likelihood function of BT model. But we do not need to be restricted to it for combining base classifiers. Let us use the α-divergence defined as follows [2] which includes KL divergence as special cases.

$$\begin{aligned}&D_{\alpha,y}[p;q] = \frac{4}{1-\alpha^2}\big(1 - \sum_{y} p(y)^{\frac{1-\alpha}{2}} q(y)^{\frac{1+\alpha}{2}}\big), \quad -1 < \alpha < 1, \\ &D_{-1,y}[p;q] = D_{KL,y}[p;q], \quad D_{1,y}[p;q] = D_{KL,y}[q;p].\end{aligned}$$

Note that α-divergence is nonnegative and is 0 iff $p(y) = q(y)$, $\forall y$. Now, let us consider minimizing

$$\mathcal{D}_\alpha(\boldsymbol{p}_{|\boldsymbol{x}}) = \sum_{a=1}^{M} D_{\alpha,a}[r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a); q_{a|\boldsymbol{x}}^{+}].$$

The cost function in eq.(1) is a special case $\mathcal{D}_{-1}(\boldsymbol{p}_{|\boldsymbol{x}})$. We can derive a new algorithm easily.

Algorithm 3

1. *initialize* $\boldsymbol{p}_{|\boldsymbol{x}}^{0}$
2. *repeat the following by increasing* $t = 0, 1, \cdots$, *until convergence*
   (a) *compute* $\boldsymbol{u}_{|\boldsymbol{x}}^{a,t}$, $a = 1, \cdots, M$,
   $$\boldsymbol{u}_{|\boldsymbol{x}}^{a,t} = \operatorname*{argmin}_{\boldsymbol{u}_{|\boldsymbol{x}}\in\mathcal{M}_a} D_{\alpha,y}[\boldsymbol{u}_{|\boldsymbol{x}}; \boldsymbol{p}_{|\boldsymbol{x}}^{t}].$$
   (b) *compute* $\boldsymbol{p}_{|\boldsymbol{x}}^{t+1}$ *which minimizes*
   $$\boldsymbol{p}_{|\boldsymbol{x}}^{t+1} = \operatorname*{argmin}_{p_{|\boldsymbol{x}}\in\mathcal{S}} \sum_{a=1}^{M} D_{\alpha,y}[\boldsymbol{u}_{|\boldsymbol{x}}^{a,t}; \boldsymbol{p}_{|\boldsymbol{x}}].$$

It is easy to check that the minimization of step 2.(a), results in (3) for every α and $D_{\alpha,y}[\boldsymbol{u}^{a,t}_{|\boldsymbol{x}}; \boldsymbol{p}^t_{|\boldsymbol{x}}] = D_{\alpha,a}[r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a); q^{+,t}_{a|\boldsymbol{x}}]$. The minimizer of step 2.(b) is also simply calculated which results in

$$
\begin{aligned}
p^{t+1}_{y|\boldsymbol{x}} &= \frac{1}{C}\Big(\sum_{a=1}^{M}\big(u^{a,t}_{y|\boldsymbol{x}}\big)^{\frac{1-\alpha}{2}}\Big)^{\frac{2}{1-\alpha}}, \quad C = \sum_{y=1}^{K}\Big(\sum_{a=1}^{M}\big(u^{a,t}_{y|\boldsymbol{x}}\big)^{\frac{1-\alpha}{2}}\Big)^{\frac{2}{1-\alpha}}, \quad -1 \le \alpha < 1 \\
p^{t+1}_{y|\boldsymbol{x}} &= \frac{1}{C}\Big(\prod_{a=1}^{M} u^{a,t}_{y|\boldsymbol{x}}\Big)^{\frac{1}{M}}, \quad C = \sum_{y=1}^{K}\Big(\prod_{a=1}^{M} u^{a,t}_{y|\boldsymbol{x}}\Big)^{\frac{1}{M}}, \quad \alpha = 1.
\end{aligned}
\tag{5}
$$

$\mathcal{D}_\alpha(\boldsymbol{p}_{|\boldsymbol{x}})$ decreases at each step and converges to a unique minimum as Algorithm 2. Resulting $\boldsymbol{p}_{|\boldsymbol{x}}$ depends on α, and how to choose α is one of our future works.

3.3 Maximum Likelihood Estimation of Base Classifiers

So far, we have discussed how to combine base classifiers $\{r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a)\}$ "when they are available." Let us denote the combined estimate based on $\mathcal{D}_\alpha(\boldsymbol{p}_{|\boldsymbol{x}})$ as $p_\alpha(y|\boldsymbol{x})$. Since it is a unique function of $\{r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a)\}$ we can rewrite it as $p_\alpha(y|\boldsymbol{x}; \{\boldsymbol{\theta}_a\})$, which shows it is a function of $\{\boldsymbol{\theta}_a\}$.

Suppose we observe data $(\boldsymbol{X}_t, Y_t)$, $t = 1, \cdots, N$. One natural question is if we can compute the MLE of $p_\alpha(y|\boldsymbol{x}; \{\boldsymbol{\theta}_a\})$, more precisely, maximizer of the following function

$$
L(\{\boldsymbol{\theta}_a\}) = \frac{1}{N}\sum_{t=1}^{N} \log p_\alpha(Y_t|\boldsymbol{X}_t; \{\boldsymbol{\theta}_a\}). \tag{6}
$$

In most of the works of ECOC, we train each classifier separately, but here, we consider if we can train them jointly. Unfortunately it is not easy in general, since $p_\alpha(y|\boldsymbol{x}; \{\boldsymbol{\theta}_a\})$ is not an exponential nor a mixture family and the explicit form is not evident. But we found it is possible when we set $\alpha = 1$ and combining only "1-against-the rest" base classifiers, that is, $I^+_a = \{y = a\}$, $a = 1, \cdots, K$. In that case, we can compute the gradient of $L(\{\boldsymbol{\theta}_a\})$ with $\boldsymbol{\theta}_a$. We show the details below. Let $p_1(a'|\boldsymbol{x}) = p_1(y = a'|\boldsymbol{x}; \{\boldsymbol{\theta}_a\})$ and $l_1(a'|\boldsymbol{x}) = \log p_1(a'|\boldsymbol{x})$. At the convergent point of Algorithm 3, $\boldsymbol{p}^t_{y|\boldsymbol{x}} = \boldsymbol{p}^{t+1}_{y|\boldsymbol{x}}$ and from eqs.(3) and (5),

$$
\begin{aligned}
l_1(a|\boldsymbol{x}) &= \frac{1}{K}\Big[\log r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a) + \sum_{a'\neq a}\log\Big\{(1 - r_{a'|\boldsymbol{x}}(\boldsymbol{\theta}_{a'}))\frac{p_1(a|\boldsymbol{x})}{1 - p_1(a'|\boldsymbol{x})}\Big\}\Big] - \log C \\
l_1(a|\boldsymbol{x}) &+ \sum_{a'\neq a}\log\left(1 - p_1(a'|\boldsymbol{x})\right) + K\log C = \log r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a) + \sum_{a'\neq a}\log(1 - r_{a'|\boldsymbol{x}}(\boldsymbol{\theta}_{a'})).
\end{aligned}
$$

Taking the derivative with respect to $\boldsymbol{\theta}_{a'}$, we have

$$
\begin{aligned}
&\partial_{\boldsymbol{\theta}_{a'}} l_1(a|\boldsymbol{x}) - \sum_{a''\neq a}\frac{p_1(a''|\boldsymbol{x})}{1 - p_1(a''|\boldsymbol{x})}\partial_{\boldsymbol{\theta}_{a'}} l_1(a''|\boldsymbol{x}) + K\partial_{\boldsymbol{\theta}_{a'}}\log C \\
&\qquad = \begin{cases} \partial_{\boldsymbol{\theta}_{a'}}\log r_{a'|\boldsymbol{x}}(\boldsymbol{\theta}_{a'}), & a' = a \\ -\dfrac{r_{a'|\boldsymbol{x}}(\boldsymbol{\theta}_{a'})}{1 - r_{a'|\boldsymbol{x}}(\boldsymbol{\theta}_{a'})}\partial_{\boldsymbol{\theta}_{a'}}\log r_{a'|\boldsymbol{x}}(\boldsymbol{\theta}_{a'}), & a' \neq a \end{cases}
\end{aligned}
\tag{7}
$$

$$
\sum_a p_1(a|\boldsymbol{x})\partial_{\boldsymbol{\theta}_{a'}} l_1(a|\boldsymbol{x}) = 0. \tag{8}
$$

Equation (8) is derived from $\sum_a p_1(a|\boldsymbol{x}) = 1$, and we have assumed $r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a)$ is differentiable. For each parameters of $\{\boldsymbol{\theta}_a\}$, eqs.(7) and (8) provides $K+1$ equations, concerning $K+1$ functions, $\partial_{\boldsymbol{\theta}_{a'}} l_1(a|\boldsymbol{x})$, $a = 1, \cdots, K$ and $\partial_{\boldsymbol{\theta}_{a'}} \log C$. This linear system of equations can be solved for each parameter, and $\partial_{\boldsymbol{\theta}_a} l_1(y|\boldsymbol{x})$ for every $(\boldsymbol{x}, y) \in (\mathcal{X}, \mathcal{Y})$ can be computed. With this gradient, we can use the gradient descent algorithm to compute the MLE.

We show some numerical experiments with the data shown in Fig. 1. Figure 3 shows the base LRs trained separately(left) or jointly using above method (right). The log likelihood function in eq.(6) increased from -0.6137 to -0.4964. By comparing the combined boundaries in Fig. 4, we see that the boundary is closer to the optimal boundary, and each base classifier becomes different from separately trained ones.

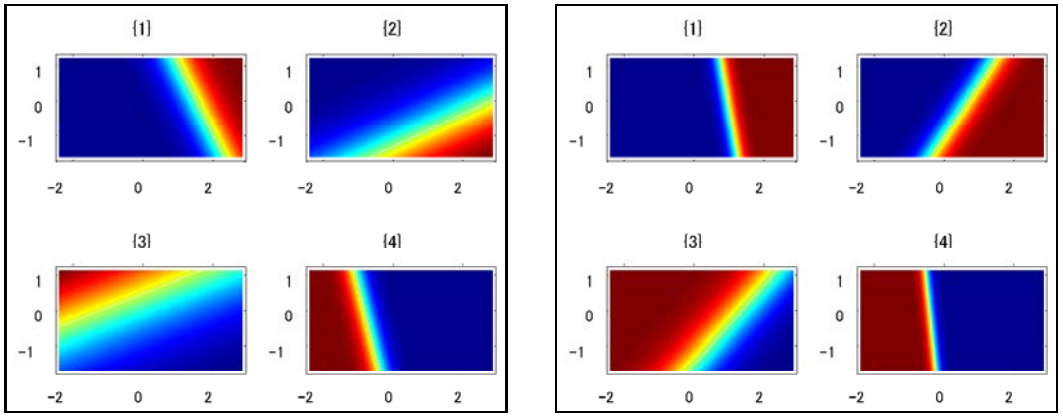


Figure 3: 4 base logistic regressions (left: trained separately, right: trained jointly)

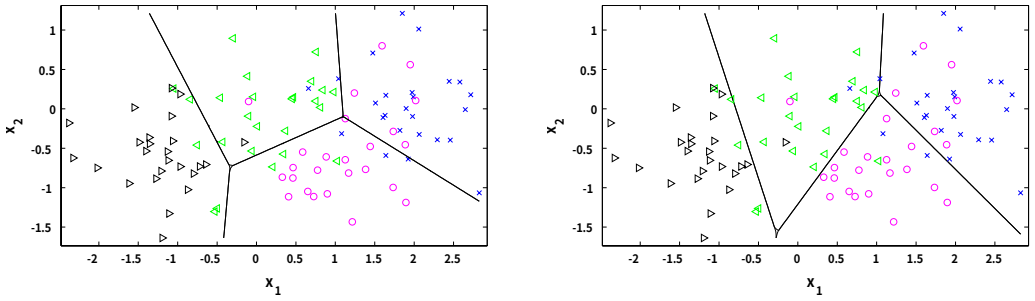


Figure 4: Decision boundary of combined estimates (left: separately trained LRs, right: jointly trained LRs)

For the base classifiers, we can use KLRs [11]. We define the base classifiers and regularized maximum

likelihood problem as follows,

$$r_{a|\boldsymbol{x}}(\theta_{a,0}, f_a) = \frac{1}{1+\exp(-f_a(\boldsymbol{x})+\theta_{a,0})},$$
$$\min_{f_a(\boldsymbol{x})} -\frac{1}{N}\sum_{t=1}^{N}\log p_1(Y_t|\boldsymbol{X}_t;\{f_a(\boldsymbol{x})\}) + \frac{\lambda}{2}\sum_a \|f_a(\boldsymbol{x})\|_{\mathcal{H}_K}. \tag{9}$$

Where $\mathcal{H}_K$ is the RKHS generated by the kernel $K(\cdot,\cdot)$. From the representer theorem, the minimizer of eq.(9) is expressed as $f_a(\boldsymbol{x}) = \sum_{t=1}^{N}\theta_{a,t}K(\boldsymbol{x},\boldsymbol{X}_t)$ and we can rewrite $r_{a|\boldsymbol{x}}(\theta_{a,0}, f_a) = r_{a|\boldsymbol{x}}(\boldsymbol{\theta}_a)$. The derivative of eq.(9) w.r.t. $\boldsymbol{\theta}_a$ can be computed by solving a linear system equations derived from eqs.(7) and (8), and $\{\boldsymbol{\theta}_a\}$ can be estimated. Figures 5 and 6 shows the results of separately trained KLRs (left) and proposed method (right). Since the original KLRs are quite efficient, there is no room for improvement, but the value of eq.(9) decreased from 0.5602 to 0.5509 by our method.

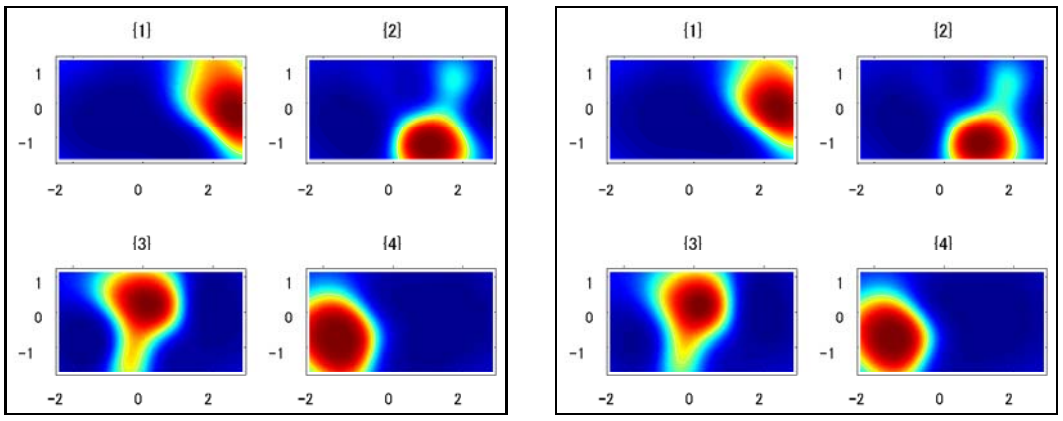


Figure 5: 4 base KLRs (left: trained separately, right: trained jointly), kernel function is $K(\boldsymbol{x},\boldsymbol{x}') = \exp(-\|\boldsymbol{x}-\boldsymbol{x}'\|^2/2\sigma^2)$ with $\sigma^2 = 0.5$, and $\lambda = 0.001$ in (9).

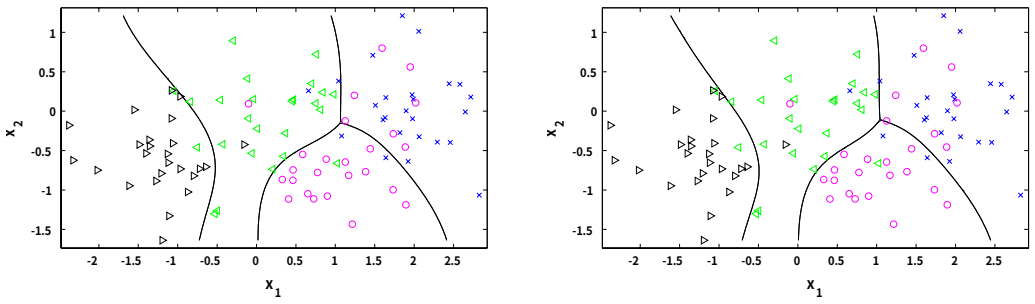


Figure 6: Decision boundary of combined estimates (left: separately trained KLRs, right: jointly trained KLRs)

4 Conclusion

Combining base classifiers is a promising idea for multi-class problems. When we use the GBT model, the combined output gives an interesting family of distribution, which is different from an exponential family nor a mixture family even if the base classifiers are simple. Two extensions were proposed in this paper, the estimation algorithm for the GBT model with the associated divergence and the base classifiers' learning algorithm.

The algorithm for the GBT model has a good property for the α–divergence. A different α will yield a different estimation and further study is necessary. Learning base classifiers is an important idea. In the framework of the error correction codes, the noisy channel is a stochastic process. In ECOC, although base classifiers gives probability as their outputs, they act in a deterministic way, and we can train them to make the combined prediction better. We have shown the gradient of the likelihood can be computed when $\alpha = 1$ and base classifiers are "1-vs-the rests." The computational cost of the gradient is rather large. We need to combine base classifiers (with Algorithm 3) and then solve a linear system of equations with $K + 1$ variables for each data point and for every parameter of every base classifier. We are now trying to apply quadratic methods for the learning, and also trying to extend the results to wider α and combination of classifiers.

References

[1] 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, 2000.

[2] S. Amari and H. Nagaoka. *Methods of Information Geometry*. AMS and Oxford Univ. Press, 2000.

[3] R. A. Bradley and M. E. Terry. Rank analysis of incomplete block designs: The method of paired comparisons. *Biometrika*, 39:324–345, 1952.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Statistical Society, Series B*, 39:1–38, 1977.

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

[6] T. Hastie and R. Tibshirani. Classification by pairwise coupling. *Ann. of Stat.*, 26(2):451–471, 1998.

[7] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001.

[8] T.-K. Huang, R. C. Weng, and C.-J. Lin. Generalized Bradley-Terry models and multi-0class probability estimates. *Journal of Machine Learning Research*, 7:85–115, 2006.

[9] R. E. Schapire. Using output codes to boost multiclass learning problems. *Proceedings of 14th International Conference on Machine Learning*, 313-321, 1997.

[10] B. Zadrozny. Reducing multiclass to binary by coupling probability estimates. *Advances in Neural Information Processing Systems*, 14:1041–1048, 2002.

[11] J. Zhu and T. Hastie. Kernel logistic regression and the import vector machine. *Journal of Computational and Graphical Statistics*, 14(1):185–205, 2005.