INVITED PAPER *Special Section on Information-Based Induction Sciences and Machine Learning*

Boosting Learning Algorithm for Pattern Recognition and Beyond

Osamu KOMORI[†a)] ***and*** **Shinto EGUCHI**[†], ***Nonmembers***

SUMMARY This paper discusses recent developments for pattern recognition focusing on boosting approach in machine learning. The statistical properties such as Bayes risk consistency for several loss functions are discussed in a probabilistic framework. There are a number of loss functions proposed for different purposes and targets. A unified derivation is given by a generator function U which naturally defines entropy, divergence and loss function. The class of U-loss functions associates with the boosting learning algorithms for the loss minimization, which includes AdaBoost and LogitBoost as a twin generated from Kullback-Leibler divergence, and the (partial) area under the ROC curve. We expand boosting to unsupervised learning, typically density estimation employing U-loss function. Finally, a future perspective in machine learning is discussed.

key words: *AUC; boosting; entropy; divergence; ROC; U-loss function; density estimation*

1. Introduction

The methodology for pattern recognition has been actively proposed and discussed in a field related with neural computation and machine learning rather than statistics in recent decades, and hence there are a vast number of new developments beyond standard discriminant analyses such as Fisher linear discriminant analysis and logistic regression, cf. [1]. In particular, boosting and support vector machine (SVM) both have got large popularity to break through conventional methods, see [2], [3]. Statistical considerations give reasonable understandings for the performance of these methods in the community of statistics. Presently boosting has been well established as in [4] where boosting is discussed as the approximation to additive modeling on the logistic scale using maximum Bernoulli likelihood.

In this paper we put stress on the characteristic such that the boosting is not simply a single method to directly construct a discriminant function, but a hyper-method to combine selected weak classifiers. In each iteration step the learning algorithm selects the best candidate in a given dictionary of weak classifiers to linearly combine the candidate and the discriminant function. Such an idea is creative and progressive in the research of pattern recognition which incorporates a rule of majority vote with effective weights. It is noted that the performance of boosting depends on the choice of dictionary in the sense that larger dictionary gives higher approximation for the Bayes rule associated with the underlying density function, but is apt to be over-learning. Thus we have to carefully investigate the trade-off in the choice of the dictionary.

Boosting satisfies a great applicability for minimization of various loss functions. A class of U-loss functions is discussed with a close association with U-entropy and U-divergence [5], [6], where U is a generator function on the real line such as an exponential function. Any U-loss function can employ the idea of boosting with a simple change from AdaBoost. If U is monotone increasing and convex, then the classifier derived by the minimization of U-loss function is shown to satisfy Bayes risk consistency in a general probabilistic framework. It is discussed that a specific choice of U leads to the robustness for outlying in both the spaces of feature vectors and class labels [7]. Although it is not a convex function, the Heaviside function leads to an important objective function called the area under the ROC curve (AUC).

At the end, we discuss an extension of boosting for pattern recognition to other statistical analyses such as density estimation. In principle, we can define U-loss function in a situation where the probabilistic framework and the discriminant function are given, so that any statistical analyses are applicable for the boosting method. In this sense the kernel method is also applicable. There remain a lot of undeveloped areas for data analysis in machine learning. We will discuss such perspectives from the point of loss functions.

2. Boosting for Pattern Recognition

Statistical pattern recognition aims to conduct good prediction for a category of an observed variable based on a given empirical examples. This can be said to be a mathematical expression in which a human brain makes prediction for a future event based on his own experiences. In fact, the brain acquires prediction capability in a process of learning from several experiences accompanying the achievements of motor ability and language function. We need to take a careful attention to this characteristic in the discussion of the statistical pattern recognition, in particular to over-learning for the training data. The framework is given in a simple form composed of a feature vector $\boldsymbol{x}$ and a class label y, in which a mapping h of $\boldsymbol{x}$ into y is called a classifier, or classification machine. The objective is to build up the classifier h with good performance for the pattern recognition in the statistical sense.

A boosting method does not directly give any specific

Manuscript received December 31, 2010.
Manuscript revised April 24, 2011.

[†]The authors are with the Institute of Statistical Mathematics, Tachikawa-shi, 190–8562 Japan.
a) E-mail: komori@ism.ac.jp

DOI: 10.1587/transinf.E94.D.1863

proposal for a classifier, but gives a procedure combining several weak classifiers in a given set, say $\mathcal{D} = \{h_\omega : \omega \in \Omega\}$, where Ω is a parameter space. The learning algorithm implements in a reasonable way a convex combination based on a training data of n tuple examples, say $D = \{(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_n, y_n)\}$, so that a strong classifier is integrated to outperform all the weak classifiers in the set. In this combination process we employ the training data many times to select weak classifiers, in which we can see the number of examples $(\boldsymbol{x}_i, y_i)$ that a weak classifier h wrongly predicts, that is $h(\boldsymbol{x}_i) \neq y_i$ for all the iteration steps.

A simple way of reweighting to the training data efficiently works to take a weighted majority vote of the sequence of weak classifiers. On the other hand, SVM is a batch-type learning algorithm to maximize the margin associated with D by the use of mathematical programming, which can sophisticatedly employ kernel functions to produce a linear decision boundary in the reproducing kernel Hilbert space. Thus SVM leads to an effective classifier associated with a higher-dimension space other than the original feature space.

There are various applications in pattern recognition since the first application to Fisher's iris data, in which a decision maker wants to predict a categorical variable, or phenotype from a given input variable, or feature variable. For example, the class label represents an endpoint in a context of risk analysis.

2.1 U-Boosting

For a training data set $D = \{(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_n, y_n)\}$, we discuss a leaning algorithm as follows. Consider a discriminant function $F : \mathcal{X} \to \mathbb{R}$ to construct a classifier $h : \boldsymbol{x} \mapsto y$ with the relation that $h(\boldsymbol{x}) = \operatorname{sgn} F(\boldsymbol{x})$, where $\mathcal{X}$ is a feature space, sgn denotes the sign function. We prepare a dictionary of weak classifiers $\mathcal{D} = \{h(\boldsymbol{x}, \omega) : \omega \in \Omega\}$ which is assumed to be the negation-closed, that is, if $h \in \mathcal{D}$, then $-h \in \mathcal{D}$. For example, the class of all linear classifier $\mathcal{D} = \{\operatorname{sgn} l(\boldsymbol{x}) : l \in \mathcal{L}\}$ with the class $\mathcal{L}$ of linear functions of $\boldsymbol{x}$ can be considered.

Let $U : \mathbb{R} \to \mathbb{R}$ be a convex and monotone increasing function. Then we define U-loss function for the discriminant function F by

$$L_U(F) = \frac{1}{n} \sum_{i=1}^{n} U(-y_i F(\boldsymbol{x}_i)), \tag{1}$$

in which the expected loss is given by $\mathbb{L}_U(F) = \mathbb{E}U(-YF(\boldsymbol{X}))$, where $\mathbb{E}$ denotes the statistical expectation of the underlying distribution for D. Our proposal is to find

$$F_U = \operatorname*{argmax}_{F \in \operatorname{con}(\mathcal{D})} \mathbb{L}_U(F),$$

where $\operatorname{con}(\mathcal{D})$ is the cone of $\mathcal{D}$, that is,

$$\operatorname{con}(\mathcal{D}) = \{\alpha_1 h_1 + \alpha_2 h_2 : \alpha_1, \alpha_2 \in \mathbb{R}^+, h_1, h_2 \in \mathcal{D}\}.$$

A variational argument leads to

$$\frac{p(y=+1|\boldsymbol{x})}{p(y=-1|\boldsymbol{x})} = \frac{\dot{U}(F_U(\boldsymbol{x}))}{\dot{U}(-F_U(\boldsymbol{x}))},$$

which implies the Bayes risk consistency such that

$$F_U(\boldsymbol{x}) = \Psi^{-1}(p(y=+1|\boldsymbol{x})),$$

where $\Psi(f) = \dot{U}(f)/\{\dot{U}(f) + \dot{U}(-f)\}$. Note that there exists the inverse function of Ψ since

$$\frac{\partial}{\partial f}\Psi(f) = \frac{\ddot{U}(f)\dot{U}(-f) + \ddot{U}(-f)\dot{U}(f)}{\{\dot{U}(f) + \dot{U}(-f)\}^2} > 0$$

from the assumption of U.

On the other hand, the U-loss function has a normalized form defined by

$$\begin{aligned} L_U(F) = \frac{1}{n} \sum_{i=1}^{n} \{ & - y_i F(\boldsymbol{x}_i) \\ & + U(F(\boldsymbol{x}_i) - b(\boldsymbol{x}_i)) + U(-F(\boldsymbol{x}_i) - b(\boldsymbol{x}_i))\}, \end{aligned} \tag{2}$$

where $b(\boldsymbol{x})$ is the normalizing factor satisfying

$$\dot{U}(F(\boldsymbol{x}) - b(\boldsymbol{x})) + \dot{U}(-F(\boldsymbol{x}) - b(\boldsymbol{x})) = 1.$$

In this way we have two forms of loss functions as in (1) and (2). If $U(t) = \exp(t)$, then (1) is exp-loss, and (2) is log-loss

$$L_U(F) = \frac{1}{n} \sum_{i=1}^{n} \log\left(1 + \exp(-y_i F(\boldsymbol{x}_i))\right)$$

because $b(\boldsymbol{x}) = \log\{\exp(F(\boldsymbol{x})) + \exp(-F(\boldsymbol{x}))\}$. In general $U(t) = \exp(t)$ generates the Kullback-Leibler divergence in which AdaBoost and LogitBoost are viewed as twin in this context. In a subsequent discussion we will consider the U-loss function for supervised learning.

2.2 U-Boost Algorithm

The learning algorithm for a sequential minimization of U-loss function in the convex hull of the dictionary $\mathcal{D}$ is as follows.

1. In the initial step we set $F_0(\boldsymbol{x}) = 0$ for all $\boldsymbol{x}$ in $\mathcal{X}$.
2. For t, $0 \leq t \leq T$ update as $F_{t+1}(\boldsymbol{x}) = F_t(\boldsymbol{x}) + \alpha_t h_t(\boldsymbol{x})$, where
$$(\alpha_t, h_t) = \operatorname*{argmin}_{(\alpha,h) \in \mathbb{R}^+ \times \mathcal{D}} L_U(F_t + \alpha h)$$
3. In the final, output a discriminant function as
$$F(\boldsymbol{x}) = \sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x}).$$

The main step 2 is sometimes changed to a gradient-type algorithm

$$h_t = \operatorname*{argmin}_{h \in \mathcal{D}} \frac{\partial}{\partial \alpha} L_U(F_t + \alpha h)\Big|_{\alpha=0}$$

and

$$\alpha_t = \operatorname*{argmin}_{\alpha\in\mathbb{R}} L_U(F_t + \alpha h_t).$$

In particular, this change is recommended when the cost of joint optimization in step 2 is considerable. A overlearning of this algorithm to the data set D is reported when the dictionary $\mathcal{D}$ is unbalanced with D. In fact, after only a few step, the error rate becomes 0, and any further steps do not improve the performance. In such a situation it is better to fix a predetermined sequence of step lengths independent of D, see the early stopping rule in [8]. Hence if we write the sequence by α, then the algorithm selects only the best candidate as

$$h_t = \operatorname*{argmin}_{h\in\mathcal{D}} L_U(F_t + \alpha\, h).$$

3. Boosting AUC

The expected U-loss function $\mathbb{L}_U(F) = \mathbb{E}U(-YF(\boldsymbol{X}))$ is expressed using joint probability in the same way as error rate. It is very common and useful for measuring the accuracy of classification performance. However, in medical and biological sciences, the type I error and type II error must be treated differently. Suppose a classification problem for disease screening in which the prevalence rate is very low. In that case, classifying all subjects to be negative (non-diseased) leads to almost perfect classification based on U-loss or error rate, though it is not practical. In this context, the false positive rate (FPR) and true positive rate (TPR) are used in practical situations, and the classification performance is often measured by the area under the ROC curve (AUC). See the relationship between U-loss function and the AUC in the logistic-type context [9].

3.1 Area under the ROC Curve

For probability density function $g_-(\boldsymbol{x})$ and $g_+(\boldsymbol{x})$ for $y \in \{-1, +1\}$, the FPR and TPR are defined as

$$\mathrm{FPR}(c) = \int_{F(\boldsymbol{x})\ge c} g_-(\boldsymbol{x})d\boldsymbol{x}, \text{ and } \mathrm{TPR}(c) = \int_{F(\boldsymbol{x})\ge c} g_+(\boldsymbol{x})d\boldsymbol{x},$$

where the subject is classified to be positive when $F(\boldsymbol{x}) > c$, and to be negative when otherwise. Hence we have

$$\mathrm{ROC}(F) = \{(\mathrm{FPR}(c), \mathrm{TPR}(c)) \,|c \in \mathbb{R}\}.$$

Then, the area under the ROC curve (AUC) is given as

$$\mathrm{AUC}(F) = \int_{\infty}^{-\infty} \mathrm{TPR}(c)d\mathrm{FPR}(c).$$

It is rewritten as

$$\mathrm{AUC}(F) = \int\int \mathrm{H}(F(\boldsymbol{x}_+) - F(\boldsymbol{x}_-))g_-(\boldsymbol{x}_-)g_+(\boldsymbol{x}_+)d\boldsymbol{x}_-d\boldsymbol{x}_+, \tag{3}$$

where $\mathrm{H}(z)$ is the Heaviside function: $\mathrm{H}(z) = 1$ if $z \ge 0$ and 0 otherwise. Hence, the empirical AUC is given by

$$\overline{\mathrm{AUC}}(F) = \frac{1}{n_-n_+}\sum_{i=1}^{n_-}\sum_{j=1}^{n_+} \mathrm{H}(F(\boldsymbol{x}_{+j}) - F(\boldsymbol{x}_{-i})),$$

where $\{\boldsymbol{x}_{-1}, \ldots, \boldsymbol{x}_{-n_-}\}$ and $\{\boldsymbol{x}_{+1}, \ldots, \boldsymbol{x}_{+n_+}\}$ are samples with sample size n_- and n_+ for $y = -1$ and $y = +1$, respectively. Its probabilistic interpretation is given in [10] as

$$\mathrm{AUC}(F) = P(F(\boldsymbol{X}_+) \ge F(\boldsymbol{X}_-)).$$

In order to facilitate the maximization process, the standard normal distribution function is used in place of $\mathrm{H}(z)$ [9], or a sigmoid approximation for this purpose is also proposed in [11] and [12]. Here, we consider the former approximation:

$$\overline{\mathrm{AUC}}_\sigma(F) = \frac{1}{n_-n_+}\sum_{i=1}^{n_-}\sum_{j=1}^{n_+} \mathrm{H}_\sigma(F(\boldsymbol{x}_{+j}) - F(\boldsymbol{x}_{-i})),$$

where $\mathrm{H}_\sigma(z) = \Phi(z/\sigma)$, with Φ being the standard normal distribution function.

Similarly to Eq. (3), the approximate AUC is given as

$$\mathrm{AUC}_\sigma(F) = \int\int \mathrm{H}_\sigma(F(\boldsymbol{x}_+) - F(\boldsymbol{x}_-))g_-(\boldsymbol{x}_-)g_+(\boldsymbol{x}_+)d\boldsymbol{x}_-d\boldsymbol{x}_+. \tag{4}$$

The next theorem in [13] justifies the use of the approximate AUC in place of the AUC as follows.

Theorem 1: Let

$$\Psi(\gamma) = \mathrm{AUC}_\sigma\Big(F + \gamma\, m(\Lambda)\Big),$$

where $\Lambda(\boldsymbol{x}) = g_+(\boldsymbol{x})/g_-(\boldsymbol{x})$ and m is a strictly increasing function. Then, $\Psi(\gamma)$ is a strictly increasing function of $\gamma \in \mathbb{R}$, and

$$\sup_F \mathrm{AUC}_\sigma(F) = \lim_{\gamma\to\infty}\Psi(\gamma) = \mathrm{AUC}(\Lambda).$$

Theorem 1 can be extended into the justification of the use of the approximate pAUC [14]. See Theorem 2 for more details.

3.2 Objective Function

At first, we prepare a set of weak classifiers, $\mathcal{D}_k$, for each k-th component of $\boldsymbol{x} \in \mathbb{R}^p$ and combine the sets into

$$\mathcal{D} = \bigcup_{k=1}^{p} \mathcal{D}_k,$$

among which we choose weak classifiers to construct $F(\boldsymbol{x})$.

In this setting, $F(\boldsymbol{x})$ can be decomposed componentially:

$$F(\boldsymbol{x}) = F_1(x_1) + \cdots + F_p(x_p).$$

Then, the objective function is given as

$\overline{\mathrm{AUC}}_{\sigma,\lambda}(F)$

$$= \frac{1}{n_- n_+} \sum_{i=1}^{n_-} \sum_{j=1}^{n_+} \mathrm{H}_\sigma(F(\boldsymbol{x}_{+j}) - F(\boldsymbol{x}_{-i})) - \lambda \sum_{k=1}^{p} \sum_{x_k \in \mathcal{B}_k} \left\{F_k^{(2)}(x_k)\right\}^2,$$

where λ is a smoothing parameter and $F_k^{(2)}(x_k)$ denotes the second-order difference of $F_k(x_k)$. The second-order difference is considered for $\mathcal{B}_k$, which is a set of quantiles for x_k.

By a simple calculation, we have

$$\overline{\mathrm{AUC}}_{\sigma,\lambda}(F) = \overline{\mathrm{AUC}}_{\sigma',\lambda'}\left(\frac{\sigma'}{\sigma}F\right),$$

if $\lambda\sigma^2 = \lambda'\sigma'^2$. This implies that the maximization of $\overline{\mathrm{AUC}}_{\sigma,\lambda}(F)$ is equivalent to that of $\overline{\mathrm{AUC}}_{1,\lambda\sigma^2}\left(\frac{F}{\sigma}\right)$. Therefore, we have

$$\max_{\sigma,\lambda,F} \overline{\mathrm{AUC}}_{\sigma,\lambda}(F) = \max_{\lambda,F} \overline{\mathrm{AUC}}_{1,\lambda}(F).$$

From this consideration, we can fix $\sigma = 1$ without loss of generality, and redefine $\overline{\mathrm{AUC}}_\lambda(F) \equiv \overline{\mathrm{AUC}}_{1,\lambda}(F)$.

3.3 AUCBoost Algorithm

1. Start with a discriminant function $F_0(\boldsymbol{x})$.
2. For $t = 1, \ldots, T$
   a. Find the best weak classifier h_t and calculate the coefficient α_t as
   $$h_t = \operatorname*{argmax}_{h\in\mathcal{D}} \frac{\partial}{\partial\alpha}\overline{\mathrm{AUC}}_\lambda(F_{t-1} + \alpha h)\Big|_{\alpha=0},$$
   $$\alpha_t = \operatorname*{argmax}_{\alpha>0} \overline{\mathrm{AUC}}_\lambda(F_{t-1} + \alpha h_t).$$
   b. Update the discriminant function as
   $$F_t(\boldsymbol{x}) = F_{t-1}(\boldsymbol{x}) + \alpha_t h_t(\boldsymbol{x}).$$
3. Finally, output the final discriminant function:
   $$F(\boldsymbol{x}) = F_0(\boldsymbol{x}) + \sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x}).$$

If we have no prior information about the data, we set $F_0(\boldsymbol{x}) = 0$. In step 2.a, we search $\mathcal{D}$ for a h_t which maximizes the first derivative of $\overline{\mathrm{AUC}}_\lambda(F)$ at the point $F_{t-1}(\boldsymbol{x}) + \alpha h(\boldsymbol{x})$. This argument is similar to that of [3] and [7]. Next, we calculate the coefficient of $h_t(\boldsymbol{x})$ using the Newton-Raphson method, and add $\alpha_t h_t(\boldsymbol{x})$ to the previous discriminant function. We repeat this process T times and output the final discriminant function. Thus, the resultant discriminant function is an aggregation of $h_t(\boldsymbol{x})$'s with weights α_t's.

4. Boosting pAUC

In medical practice, a part of the range of FPR or TPR is essential. For example, in disease screening, the targeted population consists mainly of healthy subjects. In that case, a very low FPR is required to avoid a large amount of unnecessary treatments. On the other hand, in the case where severe medical treatments such as biopsies or surgeries follow the diagnosis of subjects when being judged to be positive, TPR needs to be kept as high as possible. In this context, the partial area under the ROC curve is getting more useful than the AUC itself. The classification problems relating to the pAUC are discussed in several papers such as [14]–[16].

4.1 Partial Area under the ROC Curve

We consider a part of the AUC by limiting the value of FPR between α_1 and α_2, which are determined by thresholds c_1 and c_2, respectively:

$$\alpha_1 = \int \mathrm{H}(F(\boldsymbol{x}) - c_1) g_-(\boldsymbol{x}) d\boldsymbol{x},\ \alpha_2 = \int \mathrm{H}(F(\boldsymbol{x}) - c_2) g_-(\boldsymbol{x}) d\boldsymbol{x}, \tag{5}$$

where $0 \le \alpha_1 < \alpha_2 \le 1$ ($c_2 < c_1$). Usually, the values are set to be 0 and 0.1, respectively. However, it is also worth considering to take $\alpha_1 > 0$ and choose $\alpha_2 - \alpha_1$ to be small enough, so that we essentially maximize TPR for fixed FPR. Then, the pAUC can be divided into a fan-shaped part and a rectangular part:

$$\begin{aligned}
&\mathrm{pAUC}(F, \alpha_1, \alpha_2) \\
&\quad = \int_{c_1}^{c_2} \mathrm{TPR}(c) d\mathrm{FPR}(c) \\
&\quad = \int_{c_1}^{c_2} \int_{c_2 \le F(\boldsymbol{x}) \le c_1} \mathrm{H}(F(\boldsymbol{x}) - c) g_+(\boldsymbol{x}) d\boldsymbol{x} d\mathrm{FPR}(c) \\
&\qquad + \mathrm{TPR}(c_1)(\alpha_2 - \alpha_1).
\end{aligned}$$

Its probabilistic interpretation is offered by [17] as

$$\mathrm{pAUC}(F, \alpha_1, \alpha_2) = P(F(\boldsymbol{X}_+) \ge F(\boldsymbol{X}_-),\ c_2 \le F(\boldsymbol{X}_-) \le c_1).$$

The empirical form is expressed as

$$\overline{\mathrm{pAUC}}(F, \overline{\alpha}_1, \overline{\alpha}_2) = \frac{1}{n_- n_+} \sum_{i\in I} \sum_{j=1}^{n_+} \mathrm{H}(F(\boldsymbol{x}_{+j}) - F(\boldsymbol{x}_{-i})),$$

where $\overline{\alpha}_1$ and $\overline{\alpha}_2$ are empirical values that are the closest to α_1 and α_2, respectively; $I = \{i |\ \overline{c}_2 \le F(\boldsymbol{x}_{-i}) \le \overline{c}_1\}$, where $\overline{c}_1$ and $\overline{c}_2$ are thresholds determined by $\overline{\alpha}_1$ and $\overline{\alpha}_2$.

In the same way as Eq. (4), the approximate pAUC is given as

$$\begin{aligned}
&\mathrm{pAUC}_\sigma(F, \alpha_1, \alpha_2) \\
&\quad = \int_{c_1}^{c_2} \int_{c_2 \le F(\boldsymbol{x}) \le c_1} \mathrm{H}_\sigma(F(\boldsymbol{x}) - c) g_+(\boldsymbol{x}) d\boldsymbol{x} d\mathrm{FPR}(c) \\
&\qquad + \mathrm{TPR}(c_1)(\alpha_2 - \alpha_1),
\end{aligned}$$

where α_1 and α_2 are defined in (5). Similarly, the corresponding empirical pAUC is defined as

$$\overline{\mathrm{pAUC}}_\sigma(F, \overline{\alpha}_1, \overline{\alpha}_2)$$

$$= \frac{1}{n_- n_+} \sum_{i \in I} \left\{ \sum_{j \in J_{\rm fan}} \mathrm{H}_\sigma(F(\boldsymbol{x}_{+j}) - F(\boldsymbol{x}_{-i})) + \sum_{j \in J_{\rm rec}} \mathrm{H}(F(\boldsymbol{x}_{+j}) - F(\boldsymbol{x}_{-i})) \right\},$$

where $J_{\rm fan} = \{j |\ \overline{c}_2 \le F(\boldsymbol{x}_{+j}) \le \overline{c}_1\}$ and $J_{\rm rec} = \{j |\ \overline{c}_1 < F(\boldsymbol{x}_{+j})\}$. Before discussing a boosting method for the pAUC, we give a theoretical justification of the use of the approximate pAUC in the following theorem [14].

Theorem 2: For a pair of fixed α_1 and α_2, let

$$\Psi(\gamma) = \mathrm{pAUC}_\sigma\Big(F + \gamma\, m(\Lambda), \alpha_1, \alpha_2\Big),$$

where γ is a scalar, $\Lambda(\boldsymbol{x}) = g_+(\boldsymbol{x})/g_-(\boldsymbol{x})$ and m is a strictly increasing function. Then, $\Psi(\gamma)$ is a strictly increasing function of γ, and

$$\sup_F \mathrm{pAUC}_\sigma(F, \alpha_1, \alpha_2) = \lim_{\gamma \to \infty} \Psi(\gamma) = \mathrm{pAUC}(\Lambda, \alpha_1, \alpha_2).$$

As proved by [9] and [18], the likelihood ratio $\Lambda(x)$ is the optimal discriminant function that maximizes the AUC as well as the pAUC. Theorem 2 suggests a weak version of the Bayes risk consistency in the limiting sense.

4.2 pAUCBoost Algorithm

The difference from AUCBoost algorithm is that the thresholds $\overline{c}_1$ and $\overline{c}_2$ should be calculated, and that they depend on a discriminant function $F(\boldsymbol{x})$. Hence, the coefficient should be individually calculated for each weak classifier h, which is explicitly denoted by $\beta(h)$ in the following algorithm.

1. Start with a discriminant function $F_0(\boldsymbol{x}) = 0$ and set each coefficient $\beta_0(h)$ of weak classifiers to be 1 or -1.
2. For $t = 1, \ldots, T$
   a. Calculate the values of thresholds $\overline{c}_1$ and $\overline{c}_2$ for each $F_{t-1} + \beta_{t-1}(h)h$.
   b. Update $\beta_{t-1}(h)$ to $\beta_t(h)$ with a one-step Newton-Raphson iteration.
   c. Find the best weak classifier h_t
   $$h_t = \operatorname*{argmax}_{h \in \mathcal{D}} \overline{\mathrm{pAUC}}_\lambda(F_{t-1} + \beta_t(h)h, \overline{\alpha}_1, \overline{\alpha}_2)$$
   d. Update the discriminant function as
   $$F_t(\boldsymbol{x}) = F_{t-1}(\boldsymbol{x}) + \beta_t(h_t)h_t(\boldsymbol{x}).$$
3. Finally, output a final discriminant function $F(\boldsymbol{x}) = \sum_{t=1}^T \beta_t(h_t)h_t(\boldsymbol{x})$.

The dependency of the $\overline{\mathrm{pAUC}}_\lambda(F_{t-1} + \beta_t(h)h, \overline{\alpha}_1, \overline{\alpha}_2)$ on thresholds $\overline{c}_1$ and $\overline{c}_2$ makes it necessary to pick up the best pair of $(\beta_t(h_t), h_t)$ at the same time in step 2.c. Because of the dependency and the difficulty of getting the exact solution of $\beta_t(h_t)$, the one-step Newton-Raphson calculation is conducted in the boosting process. In this algorithm, the components $x_1, \ldots, x_p$ of $\boldsymbol{x}$ are combined componentially for maximizing the pAUC using natural cubic splines or decision stumps (single-level decision trees) in a dictionary $\mathcal{D}$, according to the values of variables (continuous or discrete). See [14] for more details.

5. Boosting for Density Estimation

A lot of boosting methods for prediction or classification have been proposed so far. The first and typical one in machine learning community is AdaBoost [19] for the minimization of the exponential loss. Other boosting methods for various objective function such as likelihood, L_2-loss, mixture of the exponential loss and naive loss, U-loss, AUC and pAUC [4], [5], [7], [13], [14], [20] have been considered and applied to real data analysis. However, the boosting methods for other purpose than prediction seem to have been paid little attention, see [21]–[23]. Recently, [24] has proposed a stagewise methods for density estimation based on L_2 loss and derived a non-asymptotic error bound. See [25] for further details. Then [26] extended the estimation method based on U-divergence and [27] modified it so that it can be applied in more general setting and with less computational cost.

5.1 U-Divergence

We employ the same generator function U to define the loss function for a density estimator. Here we redefine U as follows. Let U be a convex and monotone increasing function, and u be the first derivative. Then the conjugate convex function is given as

$$\Xi(s) = \max_{t \in \mathbb{R}} \{st - U(t)\}.$$

By differentiating it with respect to t, we have

$$\Xi(s) = s\xi(s) - U(\xi(s)),$$

where ξ is the inverse function of u. Then, for $\boldsymbol{x} \in \mathbb{R}^p$, $f(\boldsymbol{x}) > 0$ and $g(\boldsymbol{x}) > 0$, the U-divergence is defined as

$$D_U(f, g) = \int U\Big(\xi(g(\boldsymbol{x}))\Big) - U\Big(\xi(f(\boldsymbol{x}))\Big) - f(\boldsymbol{x})\{\xi(g(\boldsymbol{x})) - \xi(f(\boldsymbol{x}))\} d\boldsymbol{x}.$$

It is rewritten as

$$D_U(f, g) = C_U(f, g) - H_U(f),$$

where

$$C_U(f, g) = \int U\Big(\xi(g(\boldsymbol{x}))\Big) - f(\boldsymbol{x})\xi(g(\boldsymbol{x}))d\boldsymbol{x}$$

and

$$H_U(f) = \int U\Big(\xi(f(\boldsymbol{x}))\Big) - f(\boldsymbol{x})\xi(f(\boldsymbol{x}))d\boldsymbol{x}.$$

Here, $C_U(f,g)$ and $H_U(f)$ are U-cross entropy and U-entropy, respectively. From the relation that $H_U(f) = -\int \Xi(f(\boldsymbol{x}))d\boldsymbol{x}$, we have $D_U(f,g) \geq 0$. In the case that $U(t) = \exp(t)$, we have $u(t) = \exp(t)$ and $\xi(t) = \log(t)$, which leads to

$$D_U(f,g) = \int g(\boldsymbol{x}) - f(\boldsymbol{x}) - f(\boldsymbol{x})\{\log(g(\boldsymbol{x})) - \log(f(\boldsymbol{x}))\}d\boldsymbol{x}.$$

This is the Kullback-Leibler divergence. In the same way, if we consider

$$U(t) = \frac{1}{1+\beta}(1+\beta t)^{\frac{1+\beta}{\beta}}. \tag{6}$$

Then u and ξ are given as

$$u(t) = (1+\beta t)^{\frac{1}{\beta}}, \qquad \xi(t) = \frac{t^\beta - 1}{\beta},$$

$$D_\beta(f,g) = -\frac{1}{\beta}\int f(\boldsymbol{x})(g(\boldsymbol{x})^\beta - f(\boldsymbol{x})^\beta)d\boldsymbol{x} + \frac{1}{1+\beta}\int g(\boldsymbol{x})^{1+\beta} - f(\boldsymbol{x})^{1+\beta}d\boldsymbol{x}.$$

This is the β-divergence [28], [29]. It becomes the Kullback-Leibler divergence when $\beta \to 0$; it becomes L_2 norm when $\beta = 1$.

5.2 Loss Function for β-Divergence

For observations $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, the loss function for U-divergence is given as

$$L_U(g) = -\frac{1}{n}\sum_{i=1}^n \xi(g(\boldsymbol{x}_i)) + \int U\Big(\xi(g(\boldsymbol{x}))\Big)d\boldsymbol{x}.$$

From Eq. (6), the loss function for β-divergence is given as

$$L_\beta(g) = -\frac{1}{n\beta}\sum_{i=1}^n \{g(\boldsymbol{x}_i)^\beta - 1\} + \frac{1}{1+\beta}\int g(\boldsymbol{x})^{1+\beta}d\boldsymbol{x}.$$

This loss function is known to be robust to outliers. See [30] for the application to ICA, and [31] for that to PCA mixture.

5.3 Boosting Algorithm

For a dictionary of density function $\mathcal{D}$, the dictionary used in the boosting algorithm is defined as

$$\mathcal{D}_\beta = \Big\{\psi = \xi(\phi) \,\Big|\, \phi \in \mathcal{D}\Big\},$$

where $\xi(t) = (t^\beta - 1)/\beta$. Then, we consider the following mixture model:

$$\mathcal{M} = \left\{ u\Big(\sum_{j=1}^N p_j\psi_j(\boldsymbol{x})\Big) \,\middle|\, p_1, \ldots, p_N \geq 0, \sum_{j=1}^N p_j = 1,\ \psi_1, \ldots, \psi_N \in \mathcal{D}_\beta \right\},$$

based on which we construct the density estimator $\hat{f}$.

For a positive numerical sequence $\pi_1, \ldots, \pi_T$, the stage-wise algorithm for $\hat{f}$ is proposed by [26] as follows.

1. Choose $f_0 \in \mathcal{D}$ so that
$$L_\beta(f_0) \leq \inf_{\phi\in\mathcal{D}} L_\beta(\phi) + \epsilon,$$
where $\epsilon > 0$ is an approximation bound.
2. For $t = 1, \ldots, T$, f_t is given as
$$f_t = u\Big((1-\pi_t)\xi(f_{t-1}) + \pi_t\xi(\phi_t)\Big),$$
where, ϕ_t is chosen such that
$$L_\beta(f_t) \leq \inf_{\phi\in\mathcal{D}} L_\beta\Big(u\Big((1-\pi_t)\xi(f_{t-1}) + \pi_t\xi(\phi)\Big)\Big) + \pi_t\epsilon.$$
3. Finally, we have $\hat{f} = f_T \in \mathcal{M}$.

The numerical performance of this method is illustrated and the non-asymptotic error bound is derived in [26].

6. Discussion and Future Problems

We overview a unified perspective associated with U-loss function. In fact, any generator function U leads to a cross/diagonal entropy and divergence, in which U-cross entropy easily yields U-loss function by plugging the empirical distribution because this is a linear functional with respect to the data distribution. In this framework U model and U estimator are connected with a dualistic structure in the sense of information geometry, see [32].

Hence U-loss function naturally utilizes boosting learning by the use of prescribed set of weak classifiers, called dictionary, while U-loss function utilizes kernel methods for linear learning in the reproducing kernel Hilbert space. This tells us that such boosting and kernel methods are applicable for any loss functions such as the AUC, which is not convex but still applicable for boosting method as discussed here. In some applications we can build boosting learning algorithms for mixture model and principal/independent component analysis. AdaBoost and SVM have been established as the most popular methods in pattern recognition, however, we remark that what they have done by specific choice of the loss function is not so essential. We have not explored yet the performance of integrating local learning by specific choice of the loss function here. In the near future, it may be possible that the surprising performance is implemented for data learning in machine learning.

Acknowledgements

The authors would like to express acknowledgement to Associate Professor Kanta Naito who kindly gave us some useful comments and suggestions to this paper. We also note that this study is supported by the Program for Promotion of Fundamental Studies in Health Sciences of the National Institute of Biomedical Innovation (NIBIO).

References

[1] G.J. McLachlan, Discriminant Analysis and Statistical Pattern Recognition, Wiley & Sons, Hoboken, 2004.
[2] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, New York, 2006.
[3] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction (second edition), Springer, New York, 2009.
[4] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," The Annals of Statistics, vol.28, pp.337–407, 2000.
[5] N. Murata, T. Takenouchi, T. Kanamori, and S. Eguchi, "Information geometry of U-Boost and Bregman divergence," Neural Comput., vol.16, pp.1437–1481, 2004.
[6] S. Eguchi, "Information geometry and statistical pattern recognition," Sugaku Expositions, vol.19, pp.197–216, 2006.
[7] T. Takenouchi and S. Eguchi, "Robustifying AdaBoost by adding the naive error rate," Neural Comput., vol.16, pp.767–787, 2004.
[8] T. Zhang and B. Yu, "Boosting with early stopping: Convergence and consistency," The Annals of Statistics, vol.4, pp.1538–1579, 2005.
[9] S. Eguchi and J. Copas, "A class of logistic-type discriminant functions," Biometrika, vol.89, pp.1–22, 2002.
[10] D. Bamber, "The area above the ordinal dominance graph and the area below the receiver operating characteristic graph," J. Mathematical Psychology, vol.12, pp.387–415, 1975.
[11] S. Ma and J. Huang, "Regularized ROC method for disease classification and biomarker selection with microarray data," Bioinformatics, vol.21, pp.4356–4362, 2005.
[12] Z. Wang, Y.I. Chang, Z. Ying, L. Zhu, and Y. Yang, "A parsimonious threshold-independent pretein feature selection method through the area under receiver operating characteristic curve," Bioinformatics, vol.23, pp.2788–1794, 2007.
[13] O. Komori, "A boosting method for maximization of the area under the ROC curve," Annals of the Institute of Statistical Mathematics, 2009. (online).
[14] O. Komori and S. Eguchi, "A boosting method for maximizing the partial area under the ROC curve," BMC Bioinformatics, vol.11, p.314, 2010.
[15] M.S. Pepe and M.L. Thompson, "Combining diagnostic test results to increase accuracy," Biostatistics, vol.1, pp.123–140, 2000.
[16] Z. Wang and Y.I. Chang, "Markers selection via maximizing the partial area unber the ROC curve of linear risk scores," Biostatistics, vol.12, pp.369–385, 2011.
[17] M.S. Pepe, The Statistical Evaluation of Medical Tests for Classification and Prediction, Oxford University Press, New York, 2003.
[18] M.W. McIntosh and M.S. Pepe, "Combining several screening tests: Optimality of the risk score," Biometrics, vol.58, pp.657–664, 2002.
[19] Y. Freund and R.E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," J. Computer and System Sciences, vol.55, pp.119–139, 1997.
[20] G. Tutz and H. Binder, "Generalized additive modeling with implicit variable selection by likelihood-based boosting," Biometrics, vol.62, pp.961–971, 2006.
[21] G. Ridgeway, "Looking for lumps: Boosting and bagging for density estimation," Computational Statistics & Data Analysis, vol.38, pp.379–392, 2002.
[22] S. Rosset and E. Segal, "Boosting density estimation," Advances in Neural Information Processing System 15, 2003.
[23] M.D. Marzio and C.C. Taylor, "On boosting kernel density methods for multivariate data: Density estimation and classification," Statistical Methods & Applications, vol.14, pp.163–178, 2005.
[24] J. Klemelä, "Density estimation with stagewise optimization of the empirical risk," Mach. Learn., vol.67, pp.169–195, 2007.
[25] J. Klemelä, Smoothing of Multivariate Data, Density Estimation and Visualization, John Wiley & Sons, Hoboken, New Jersey, 2009.
[26] K. Naito and S. Eguchi, "Density estimation with minimization of U-divergence," submitted, 2010.
[27] O. Komori, K. Naito, and S. Eguchi, "Boosting for density estimation based on U loss function," IEICE Technical Report, IBISML2010-69, 2010.
[28] A. Basu, I.R. Harris, N. Hjort, and M. Jones, "Robust and efficient estimation by minimizing a density power divergence," Biometrika, vol.85, pp.549–559, 1998.
[29] M. Minami and S. Eguchi, "Robust blind source separation by beta divergence," Neural Comput., vol.14, pp.1859–1886, 2002.
[30] M.N.H. Mollah, M. Minami, and S. Eguchi, "Exploring latent structure of mixture ica models by the minimum beta-divergence method," Neural Comput., vol.18, pp.166–190, 2006.
[31] M.N.H. Mollah, N. Sultana, M. Minami, and S. Eguchi, "Robust extraction of local structures by the minimum beta-divergence method," Neural Netw., vol.23, pp.226–238, 2010.
[32] F. Emmert-Streib and M. Dehmer, Information Theory and Statistical Learning, Springer, New York, 2009.

Osamu Komori got Ph.D degree at The Graduate University for Advanced Studies. He is a project researcher at Prediction and Knowledge Discovery Research Center in The Institute of Statistical Mathematics.

Shinto Eguchi got Ph.D degree at Hiroshima University. He is a professor and the chief of Prediction and Knowledge Discovery Research Center in The Institute of Statistical Mathematics.