Research Memorandum No. 1006 September 13, 2006

A bridge between boosting and a kernel machine

Masanori KAWAKITA
Shiro IKEDA
and
Shinto EGUCHI

The Institute of
Statistical Mathematics

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

A bridge between boosting and a kernel machine

Masanori Kawakita kawakita@ism.ac.jp
Shiro Ikeda shiro@ism.ac.jp
Shinto Eguchi eguchi@ism.ac.jp

Institute of Statistical Mathematics

September 13, 2006

Abstract

In this paper, boosting methods are studied from a viewpoint of kernel machines. This natural connection has already been revealed by defining a kernel function associated with the set of weak learners, which we call the WL kernel (Weak Learner kernel). We review this connection with respect to a kernel exponential family, and propose two important extensions of boosting methods for classification problems. First proposal is a new simple regularized boosting, which is confirmed to be valid through some experiments on real data. The other is a new simple kernel function from the investigation of the RKHS of decision stumps, which is one of the most widely-used weak learners. Several experiments confirm the efficiency and the validity of the proposed algorithm with the new kernel function.

1 Introduction

Both boosting and kernel methods provide attractive statistical classification tools. Boosting methods (Breiman, 1998; Freund and Schapire, 1997; Friedman et al., 2000) generally construct an accurate classifier by combining many weak learners, while the kernel methods Schölkopf and Smola (2001) map the inputs to a high dimensional feature space. As was pointed out in Rätsch et al. (2000), their natural connection becomes evident by defining a kernel function (we call the WL kernel) associated with the set of weak learners whose reproducing kernel Hilbert space (RKHS) is equal to the set of all possible discriminant functions of boosting.

In this paper, we study boosting methods from a viewpoint of the kernel exponential family (Zhu and Hastie, 2005; Canu and Smola, 2006). This viewpoint enables boosting to enjoy several results of the kernel methods. First, we demonstrate how the complexity of boosting discriminant function changes by showing the Gram matrices. Here, we employed the decision stumps, which are one of the most widely-used weak learners. Although the WL kernel works fine,

the use of the WL kernel requires much computational cost when the number of weak learners is large. To overcome this problem, we derive a new simple kernel function associated with decision stumps. The proposed kernel, referred to as a *decision stump kernel*, requires much less computational cost and has no intractable parameters. Secondly, we propose a simple regularized boosting algorithm. Different from several proposed regularized boosting algorithms (Rätsch et al., 1999; Sun et al., 2004; Rätsch et al., 2001), where regularizers are based on the weights in the boosting algorithm or on the soft margin concept, our proposal is to use the inner product defined in the RKHS induced by the WL kernel. We also give the upperbound of its generalization error. Several experiments on real data sets confirm the validity of the proposed algorithm with the WL kernel, and show the decision stump kernel also works fine.

The paper is organized as follows: Section 2 gives a brief review the boosting algorithm. In Section 3, we connect boosting to a kernel machine by introducing the WL kernel. We also derive decision stump kernel and a new regularized boosting algorithm. Section 4 shows experimental results, and section 5 concludes the paper with discussions.

2 Review of boosting algorithm

In this section, we briefly summarize AdaBoost (Freund and Schapire, 1997) with the geometrical understanding (Lebanon and Lafferty, 2002). In AdaBoost, it is assumed that we have only inaccurate classifiers referred to as *weak learners*, and AdaBoost linearly combines weak learners to construct an accurate classifier.

We consider the following binary classification problem. Let $\mathcal{X} \subset \Re^M$ be a feature space and $\mathcal{Y} = \{-1, 1\}$ be a binary label set. Let further $(X, Y) \in (\mathcal{X}, \mathcal{Y})$ be a pair of random variables generated from the joint distribution of (X, Y) denoted by $p(x, y) = q(x)p(y \,|\, x)$ where $q(x)$ denotes the underlying distribution of X and $p(y \,|\, x)$ is the underlying distribution of Y conditioned on $X = x$. The extended Kullback-Leibler (KL) divergence (Lebanon and Lafferty, 2002) between any nonnegative measure $\mu(y \,|\, x)$ and $\nu(y \,|\, x)$ (not restricted to probability density functions) is defined as

$$D(\mu, \nu) = \int_{\mathcal{X}} q(x) \sum_{y \in \mathcal{Y}} \{\nu(y \,|\, x) - \mu(y \,|\, x) - \mu(y \,|\, x)(\log(\nu(y \,|\, x)) - \log(\mu(y \,|\, x)))\} \, dx. \tag{1}$$

Let $\mathcal{C}$ be a set of all available weak learners, *i.e.*, $\mathcal{C} = \{f_j : \mathcal{X} \to \mathcal{Y} \,|\, j = 1, 2, \cdots\}$. Let further $\Phi(x) = (f_1(x), f_2(x), \cdots)^T$ and $\theta = (\theta_1, \theta_2, \cdots)^T$ be a real vector that has the same dimension with $\Phi(x)$, where T denotes transposition of a vector. Then, define an exponential model associated with $\mathcal{C}$

$$\mathcal{M}(\mathcal{C}) = \left\{ \mu(y|x; \theta) = \exp\left(-\frac{y}{2} \langle \theta, \Phi(x) \rangle - g(x; \theta) \right) \right\}, \tag{2}$$

where $\langle \cdot, \cdot \rangle$ is an Euclidean inner product and $g(x; \theta) = -\left\langle \theta, \sum_{y' \in \mathcal{Y}} \frac{y'}{2} p(y' \,|\, x) \Phi(x) \right\rangle$.

Note that the model $\mathcal{M}(\mathcal{C})$ is not an usual statistical model since $\sum_{y\in\mathcal{Y}} \mu(y \mid x;\theta) \neq 1$ in general. Consider a problem to optimize $\hat{\theta}$ such that $\mu(y \mid x;\hat{\theta})$ is the nearest point among $\mathcal{M}(\mathcal{C})$ from the underlying distribution $p(y \mid x)$ with respect to KL divergence in Eq. (1), *i.e.*, $\hat{\theta} = \mathrm{argmin}_\theta\, D(p(y \mid x), \mu(y \mid x;\theta))$. For any $\mu \in \mathcal{M}(\mathcal{C})$, the divergence is calculated as

$$\begin{aligned} D(p,\mu) &= \int_{\mathcal{X}} p(x) \sum_{y\in\mathcal{Y}} \mu(y \mid x;\theta) - P(y \mid x) \\ &\quad + P(y \mid x)(\log(\mu(y \mid x;\theta)) - \log(P(y \mid x)))dx \\ &= \int_{\mathcal{X}} p(x) \sum_{y\in\mathcal{Y}} \mu(y \mid x;\theta) + P(y \mid x)\log(\mu(y \mid x;\theta))dx + \mathrm{Const} \\ &= \int_{\mathcal{X}} q(x) \sum_{y\in\mathcal{Y}} \exp\left(-\frac{y}{2}\langle\theta,\Phi(x)\rangle - g(x;\theta)\right)dx + \mathrm{Const}, \end{aligned} \tag{3}$$

where Const is a constant with respect to θ. This follows from $E[-(Y/2)\langle\theta,\Phi(x)\rangle - g(x;\theta) \mid X = x] = 0$. since

$$\begin{aligned} \sum_{y\in\mathcal{Y}} p(y \mid x)\xi(\mu(y \mid x;\theta)) &= \sum_{y\in\mathcal{Y}} p(y \mid x)(-\frac{y}{2}\langle\theta,\Phi(x)\rangle - g(x;\theta)) \\ &= \left\langle \sum_{y\in\mathcal{Y}} -\frac{y}{2} p(y \mid x)\theta, \Phi(x)\right\rangle - g(x;\theta) \\ &= 0. \end{aligned}$$

We define an exponential loss function A by the first term in Eq. (3). , *i.e.*,

$$A(F(\cdot;\theta)) = \int_{\mathcal{X}} q(x) \sum_{y\in\mathcal{Y}} \exp(-yF(x;\theta) - g(x;\theta))dx, \tag{4}$$

where $F(x;\theta) = \langle\theta,\Phi(x)\rangle$ corresponds to a discriminant function constructed by boosting. The minimization of KL divergence from the underlying distribution P to the model $\mathcal{M}(\mathcal{C})$ is exactly equal to the minimization of the exponential loss function A with respect to θ. In practical cases, the underlying distributions $q(x)$ and $p(y \mid x)$ are unknown. Instead, when the training data $\{X_i, Y_i\}_{i=1}^n$ are given, the empirical distributions $\hat{p}(x,y) = (1/n)\sum_{i=1}^n I(x = X_i)I(y = Y_i)$ is available, where I denotes an indicator function. Assuming further that there exists only a single $y' \in \mathcal{Y}$ such that $p(y' \mid x) > 0$ for each $x \in \mathcal{X}$, we have the empirical loss function defined as

$$\widehat{A}(F(\cdot;\theta)) = \frac{1}{n}\sum_{i=1}^n \exp(-Y_i F(X_i;\theta)), \tag{5}$$

where $F(x;\theta) = \langle\theta,\Phi(x)\rangle$ corresponds to a discriminant function constructed by boosting. See Lebanon and Lafferty (2002) for the detailed derivation. The

function $\widehat{A}$ in Eq. (5) is in fact the loss function that AdaBoost minimizes iteratively with respect to θ as follows. Let $F_0(x) \equiv 0$ be an initial discriminant function. For a given current discriminant function, F_{t-1}, AdaBoost chooses a new weak learner, f_t, and its coefficient, α_t, iteratively as follows. For $t = 1, 2, \cdots, \tau$,

$$f_t \approx \operatorname*{argmin}_{f \in \mathcal{C}} \widehat{A}(F_{t-1} + \alpha f) \quad \text{for any positive } \alpha, \quad \alpha_t = \operatorname*{argmin}_{\alpha \in R} \widehat{A}(F_{t-1} + \alpha f_t), \tag{6}$$

and then the discriminant function is updated as $F_t(x) = F_{t-1}(x) + \alpha_t f_t(x)$. The final classifier is obtained as $g(x) = \mathrm{sign}(F_\tau(x;\theta))$ after τ repetitions of this process. Thus, AdaBoost can be interpreted as the iterative algorithm of minimizing empirical KL divergence from the empirical distribution $\hat{p}$ to the model $\mathcal{M}(\mathcal{C})$.

Finally, we remark that the above description about AdaBoost can be extended to a general boosting with any convex and increasing loss function (Friedman et al., 2000; Murata et al., 2004). We also remark that, by taking the bias function $g(x;\theta)$ in Eq. (2) such that $g(x;\theta) = \log \sum_{y' \in \mathcal{Y}} \exp(-\frac{y'}{2} \langle \theta, \Phi(x) \rangle)$, then the exponential model $\mathcal{M}(C)$ becomes an usual statistical model. In this case, $\widehat{A}$ reduces to an usual log-likelihood and the above algorithm reduces to LogitBoost (Friedman et al., 2000). Note that the first optimization in Eq. (6) does not depend on the value of α'. This does not exactly hold when we use another loss function but approximately holds in the sense of a one-dimensional approximation. For details, see Murata et al. (2004), for example.

3 Boosting as a kernel machine

3.1 Interpretation of boosting in the view of kernel machine

We show that boosting and a kernel machine are unified in the framework of kernel exponential family. We first introduce a kernel function $K_{\mathcal{C}}(x, x')$ associated with weak learners $\mathcal{C}$ defined as

$$K_{\mathcal{C}}(x, x') = \sum_{f_j \in \mathcal{C}} \pi_j f_j(x) f_j(x'), \tag{7}$$

where $0 \leq \pi_j \leq 1$ and $\sum_j \pi_j = 1$. Note that, if $\{\pi_j\}$ are uniform, $K_{\mathcal{C}}$ is just equal to the one suggested in Rätsch et al. (2000). The kernel function $K_{\mathcal{C}}$ can be interpreted as follows. Suppose that a random classifier $\mathbb{F}$ such that $\mathbb{F}$ is randomly chosen from $\mathcal{C}$ according to the probability $\{\pi_j\}$, *i.e.*, $\mathrm{Prob}(\mathbb{F} = f_j) = \pi_j$. When $x, x' \in \mathcal{X}$ are given, the value of $K_{\mathcal{C}}(x, x')$ is just the average of $\mathbb{F}(x)\mathbb{F}(x')$. Noting that $\mathbb{F}(x)\mathbb{F}(x') = 2I(\mathbb{F}(x) = \mathbb{F}(x')) - 1$, we further have

$$\begin{aligned} K_{\mathcal{C}}(x, x') &= E_{\mathbb{F}}[\mathbb{F}(x)\mathbb{F}(x')] = E_{\mathbb{F}}[2I(\mathbb{F}(x) = \mathbb{F}(x')) - 1] \\ &= 2E_{\mathbb{F}}[I(\mathbb{F}(x) = \mathbb{F}(x'))] - 1 = 2\mathrm{Prob}(\mathbb{F}(x) = \mathbb{F}(x')) - 1. \end{aligned}$$

Thus, the larger the probability of the event that the chosen classifier assigns the same label to x and x' is, the larger value $K_{\mathcal{C}}$ takes in the closed interval $[-1, 1]$. We refer to $K_{\mathcal{C}}$ as a *Weak Learner kernel* (WL kernel) in the sequel.

We define a reproducing kernel Hilbert space (RKHS) induced by $K_{\mathcal{C}}$ as

$$\mathcal{H}_{K_{\mathcal{C}}} = \Big\{ F(x;\theta) = \sum_j \theta_j f_j(x) \,|\, \theta_j \in R\,,\ f_j \in \mathcal{C} \ \text{ for any } j \Big\}, \tag{8}$$

equipped with an inner product defined as

$$\langle F, G \rangle_{\mathcal{H}_{K_{\mathcal{C}}}} = \sum_j \frac{1}{\pi_j} \theta_j \theta'_j,$$

for any $F(x;\theta), G(x;\theta') \in \mathcal{H}_{K_{\mathcal{C}}}$. It should be remarked that $\mathcal{H}_{K_{\mathcal{C}}}$ directly corresponds to just the set of all discriminant functions constructed by boosting. We may easily confirm the following reproducibility properties:

$$\begin{aligned} \forall x \in \mathcal{X}\,, \qquad & \langle F(\cdot;\theta), K_{\mathcal{C}}(\cdot, x) \rangle_{\mathcal{H}_{K_{\mathcal{C}}}} = F(x;\theta), \\ \forall x, x' \in \mathcal{X}\,, \qquad & \langle K_{\mathcal{C}}(\cdot, x), K_{\mathcal{C}}(\cdot, x') \rangle_{\mathcal{H}_{K_{\mathcal{C}}}} = K_{\mathcal{C}}(x, x'). \end{aligned} \tag{9}$$

By the reproducibility, we may rewrite the exponential family in Eq. (2) in terms of $K_{\mathcal{C}}$ as the kernel exponential family:

$$\mathcal{M}(\mathcal{C}) = \Big\{ \mu(y|x;\theta) = \exp\Big(-\frac{y}{2} \langle F(\cdot;\theta), K_{\mathcal{C}}(\cdot, x) \rangle_{\mathcal{H}_{K_{\mathcal{C}}}} - g(x;\theta) \Big) \Big| \, F \in \mathcal{H}_{K_{\mathcal{C}}} \Big\}, \tag{10}$$

where $g(x;\theta) = -\Big\langle F(\cdot;\theta), \sum_{y' \in \mathcal{Y}} \frac{y'}{2} p(y' \,|\, x) K_{\mathcal{C}}(\cdot, x)) \Big\rangle$. This kernel exponential family in Eq. (10) is of the same form with the one that was introduced in Canu and Smola (2006). Thus, the boosting algorithm uses the same model with kernel machines and is essentially equal to kernel machines. In contrast, supposing that a general kernel K (not restricted to the WL kernel) can be expanded as $K(x, x') = \sum_{j=1}^{\infty} \gamma_j \phi_j(x) \phi_j(x')$, the kernel machine is also interpreted as a boosting algorithm with weak learners $\mathcal{C} = \{\phi_j(x)\}_{j=1}^{\infty}$. The differences between both methods are usually loss functions (or statistical divergence) to be minimized and the optimization algorithm: boosting iteratively optimizes, while a support vector machine uses the quadratic programming, for example.

By exploiting this relationship, we give an interesting view to boosting. As the step increases, boosting iteratively adds weak learners to the discriminant function and then the discriminant function gets more complicated. We can observe this process with respect to the Gram matrix throughout the WL kernel function. To illustrate this, we introduce the *decision stump*, which is one of the most widely-used weak learners. A decision stump f^{s} is defined as:

$$f^{\mathrm{s}}(x; m, b) = \mathrm{sign}(x_m - b), \quad m = 1, 2, \cdots, M, \tag{11}$$

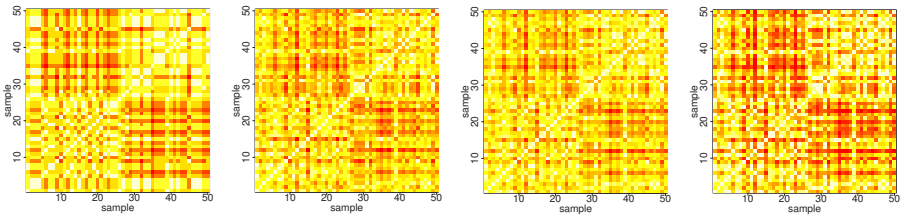
$b \in R$ is a location parameter and x_m denotes the m-th element of x. It is desirable to prepare all possible values in R as the candidate of b. However,

when a training data set $\{(X_1, Y_1), (X_2, Y_2), \cdots, (X_n, Y_n)\}$ is given, we usually prepare a finite collection of decision stumps for each feature X_m in the following manner for simplicity,

(a) Sort all unique values of the m-th feature X_m as $\{X_{i',m} \,|\, i' = 1, 2, \cdots n_m\}$ where n_m is the number of unique values of the m-th feature and $X_{i',m}$ is the m-th element of $X_{i'}$.

(b) We prepare decision stumps $f^{\mathrm{s}}(x; m, b)$ whose location parameter b are at the mid-points between sequential pairs of the sorted collection $\{X_{i',m} \,|\, i' = 1, 2, \cdots, n_m\}$.

Finally, we construct $\mathcal{C}_{\mathrm{ds}}$ by gathering all decision stumps prepared in the above steps. Therefore the number of weak learners contained in $\mathcal{C}_{\mathrm{ds}}$, is $J = \sum_{m=1}^{M}(n_m - 1)$. Using Eq. (7) with uniform distribution $\pi_j = (1/J)$, we may define a WL kernel $K_{\mathcal{C}_{\mathrm{ds}}}$ associated with decision stumps. Similarly, we also may define $K_{\mathcal{C}_{\mathrm{ds}}}$ for any subsets of $\mathcal{C}_{\mathrm{ds}}$. Panel (a) in Figure 1 shows that the WL kernel $K_{\mathcal{C}_{\mathrm{ds}}}(x, x')$ associated with weak learners of the early learning stage returns larger values if x and x' belong to the same class and vice versa. This implies the kernel machines with this $K_{\mathcal{C}_{\mathrm{ds}}}$ may work well to some extent. Panel (b) is the Gram matrix of $K_{\mathcal{C}_{\mathrm{ds}}}$ with all the decision stumps in $\mathcal{C}_{\mathrm{ds}}$. Compared to Panel (a), this has richer structure, which implies lower training error.



(a) early $K_{\mathcal{C}_{\mathrm{ds}}}$ (b) $K_{\mathcal{C}_{\mathrm{ds}}}$ (c) K_{ds} (d) K_g

Figure 1: Gram matrices on the slope data illustrated in Figure 2. The first twenty-eight samples belong to the positive class, while the remaining samples belong to the negative class. Panel (a) shows the Gram matrix of $K_{\mathcal{C}_{\mathrm{ds}}}$ consisting of the first ten weak learners chosen by AdaBoost, while Panel (b) shows the Gram matrix of all decision stumps in $\mathcal{C}_{\mathrm{ds}}$. Panel (c) shows the Gram matrix of K_{ds} (see section 3.2), while Panel (d) shows that of K_g.

3.2 Weak Learner kernel functions

In this section, we derive simple kernel functions associated with some weak learners. To obtain a good performance, it is required for boosting to prepare rich weak learners. For example, in the case of decision stumps, its location parameter b should be distributed densely in the range of each X_m, as was mentioned in the previous section. However, the naive definition of the WL

kernel in Eq. (7) requires computationally expensive calculation, as the number of weak learners increases. This is the case for decision stumps if the feature number M is large. In the following, we show that decision stumps and linear classifier leads to a simple kernel even if we prepare infinitely many of them. These new kernels are not computationally expensive.

To derive new kernels, we introduce a continuous version of $K_{\mathcal{C}}$. Assume that $\mathcal{C}$ is some statistical model that is continuously parameterized as $\mathcal{C} = \{f(x;\xi) \,|\, \xi \in \Xi\}$. Particularly to this case, we may rewrite the definition of $K_{\mathcal{C}}$ as

$$K_{\mathcal{C}}(x, x') = \int_{\xi \in \Xi} \pi(\xi) f(x;\xi) f(x';\xi) d\xi, \tag{12}$$

for any probability density function $\pi(\xi)$ on Ξ. Suppose now that each feature X_m has a finite range $[\ell_m, r_m]$, where $-\infty < \ell_m < r_m < \infty$ for each m. Redefine $\mathcal{C}_{\rm ds}$ as the set of all $f^{\rm s}$ with any b value in this range for each m, *i.e.*, $\mathcal{C}_{\rm ds} = \{f^{\rm s}(x;m,b) \,|\, m = 1, 2, \cdots, M\,, \forall m\,, b \in [\ell_m, r_m]\}$. Let the distribution of parameters of $f^{\rm s}$ be uniform, *i.e.*,

$$\pi(m,b) = \pi(m) \cdot \pi(b \,|\, m) = \frac{1}{M} \cdot \frac{1}{r_m - \ell_m}.$$

By using the definition of both Eq. (7) and (12), we define a *decision stump kernel* $K_{\rm ds}$ as

$$\begin{aligned} K_{\rm ds}(x, x') &= \sum_{m=1}^{M} \int_{\ell_m}^{r_m} \pi(m,b) f^{\rm s}(x;m,b) f^{\rm s}(x';m,b) db \\ &= \sum_{m=1}^{M} \int_{\ell_m}^{r_m} \frac{{\rm sign}(x_m - b){\rm sign}(x'_m - b)}{M(r_m - \ell_m)} db. \end{aligned}$$

This integral is easily calculated and we have

$$K_{\rm ds}(x, x') = 1 - \frac{2}{M} \sum_{m=1}^{M} \frac{|x_m - x'_m|}{r_m - \ell_m}. \tag{13}$$

Obviously, $K_{\rm ds}$ requires much less computational cost than $K_{\mathcal{C}_{\rm ds}}$ and experiments show it works as well. For instance, the Gram matrices of $K_{\mathcal{C}_{\rm ds}}$ and $K_{\rm ds}$ in Figure 1 panels (b) and (c) are quite similar. Several experiments will also confirm their similarity on their performance (Section 4).

We derive a kernel function associated with linear classifier, which is also widely-used in boosting. Let $\mathcal{C} = \{f(x;w) = w^T x \,|\, w \in \Re^M\}$. Let further $\pi(w)$ be an arbitrary distribution of the parameter w with mean 0 and covariance V. Similarly to the above calculation, $K_{\rm lin}$ is derived as

$$K_{\rm lin}(x, x') = \int_{\Re^M} \pi(w) f(x;w) f(x';w) dw = x^T \left\{ \int_{\Re^M} \pi(w) w w^T dw \right\} x' = x^T V x'.$$

This is an Euclidean inner product generalized by the matrix V, which has often appeared in linear algebra. Note that this is not a WL kernel since $f(x;w)$ is not a classifier in the exact sense.

3.3 A regularized boosting algorithm

A simple regularized boosting algorithm is proposed in this section. The algorithm will be developed based on the following proposition.

Proposition 1. *Suppose that the training data* $\{X_i, Y_i\}_{i=1}^n$ *are given. Then, there necessarily exists* $\eta \in R^n$ *such that the minimizer* $\hat{\theta}$*, i.e.,*

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \widehat{A}(F(\cdot;\theta)) + \lambda \|F(\cdot;\theta)\|^2_{\mathcal{H}_{K_\mathcal{C}}} \tag{14}$$

admits the representation of the form:

$$\hat{\theta}_j = \sum_{\ell=1}^n \pi_j f_j(X_\ell)\eta_\ell \quad \textit{for each } j.$$

The proof immediately follows from the direct application of the representer theorem (Schölkopf and Smola, 2001) with the kernel $K_\mathcal{C}$. The regularized boosting is derived by iterative minimization of the regularized loss function in Eq. (14) with respect to η as follows. We define the discriminant function that was suggested by Proposition 1 as

$$\overline{F}(x;\eta) = \sum_j \left\{ \sum_{\ell=1}^n \pi_j f_j(X_\ell)\eta_\ell \right\} f_j(x) = \sum_{\ell=1}^n \eta_\ell K_\mathcal{C}(x, X_\ell).$$

Noting that, by Eq. (9),

$$\left\langle \overline{F}, \overline{F} \right\rangle_{\mathcal{H}_{K_\mathcal{C}}} = \left\langle \sum_{\ell=1}^n \eta_\ell K_\mathcal{C}(x, X_\ell), \sum_{\ell'=1}^n \eta_{\ell'} K_\mathcal{C}(x, X_{\ell'}) \right\rangle_{\mathcal{H}_{K_\mathcal{C}}} = \sum_{\ell=1}^n \sum_{\ell'=1}^n \eta_\ell \eta_{\ell'} K_\mathcal{C}(X_\ell, X_{\ell'}),$$

we may rewrite Eq. (14) for any $\overline{F}$ as

$$\widehat{\mathbb{A}}(\overline{F}(\cdot;\eta)) = \frac{1}{n}\sum_{i=1}^n \exp(-Y_i \overline{F}(X_i;\eta)) + \lambda \eta^T \boldsymbol{K}_\mathcal{C} \eta, \tag{15}$$

where $\boldsymbol{K}_\mathcal{C}$ denotes the Gram matrix of $K_\mathcal{C}$. The algorithm is obtained in the same way with AdaBoost as was described in Eq. (6) by replacing $\widehat{A}$ with $\widehat{\mathbb{A}}$ and $\mathcal{C}$ with $\mathcal{C}' = \{(h(x) = \pm K_\mathcal{C}(x, X_\ell) \,|\, \ell = 1, 2, \cdots, n\}$. In this case, the first equation in Eq. (6) is calculated as

$$\begin{aligned}
h_t &= \underset{h\in\mathcal{C}'}{\operatorname{argmin}} \widehat{\mathbb{A}}(\overline{F}_{t-1}(\cdot;\eta^{t-1}) + \alpha h(\cdot)) \\
&\approx \underset{h\in\mathcal{C}'}{\operatorname{argmin}} \widehat{\mathbb{A}}(\overline{F}_{t-1}(\cdot;\eta^{t-1})) + \left.\frac{\partial \widehat{\mathbb{A}}(\overline{F}_{t-1}(\cdot;\eta^{t-1}) + \alpha h(\cdot))}{\partial \alpha}\right|_{\alpha=0} \alpha \\
&= \underset{h\in\mathcal{C}'}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^n \left\{ \exp(-Y_i \overline{F}_{t-1}(X_i;\eta^{t-1}))(-Y_i) + 2n\lambda\eta_i^{t-1} \right\} h(X_i)\alpha.
\end{aligned}$$

As a result, we obtain the following algorithm.

---

1. Fix a smoothing parameter $\lambda > 0$ and set η^0 as an n-dimensional zero vector.

2. For $t = 1, 2, \cdots, \tau$, repeat the following process.

   (a) Set the weights $\{D_t(i)\}$ as

   $$D_t(i) = \frac{1}{Z}\{\exp(-Y_i \overline{F}_{t-1}(X_i; \eta^{t-1}))(-Y_i) + 2n\lambda\eta_i^{t-1}\}, \qquad (16)$$

   where Z is a normalizing constant such that $\sum_i D_t(i) = 1$.

   (b) Find a new best weak learner from $\mathcal{C}'$ and its coefficient as follows:

   $$h_t = \operatorname*{argmin}_{h \in \mathcal{C}'} \sum_{i=1}^{n} D_t(i) h(X_i), \qquad (17)$$

   $$\alpha_t = \operatorname*{argmin}_{\alpha} \widehat{\mathbb{A}}(\overline{F}_{t-1}(\cdot; \eta^{t-1}) + \alpha h_t). \qquad (18)$$

   (c) Update $\{\eta_\ell^{t-1}\}$ as $\eta_\ell^t = \eta_\ell^{t-1} + \alpha_t(I(h_t = K(\cdot, X_\ell)) - I(h_t = -K(\cdot, X_\ell)))$.

3. Finally, we obtain a resultant classifier $g(x) = \operatorname{sign}(\overline{F}(x; \eta^\tau))$.

Schapire and Singer (1999) proposed the generalized version of AdaBoost, where weak learners return not the labels in $\mathcal{Y}$ but real values. Our proposal is equal to this generalized AdaBoost with weak learners $\mathcal{C}'$ except the loss function. In particular, both are exactly equal if $\lambda = 0$.

The ordinary boosting solves high dimensional optimization problem since many weak learners are often used. In contrast, the regularized boosting uses only n weak learners in $\mathcal{C}'$. As the trade-off, each weak learner of $\mathcal{C}'$ gets much more complicated than the original weak learner in $\mathcal{C}$. In a special case where decision stumps are used, however, the use of decision stump kernel K_{ds} reduces the complexity of $K_{\mathcal{C}}$ considerably, as was described in the previous section. We also remark that the above algorithm works with any kernel functions. We compare the use of decision stump kernel with the radial basis function kernel in the next section.

It is worth noting that the minimization of $\widehat{\mathbb{A}}$ can be interpreted from the Bayesian view when we restrict $\mathcal{M}(\mathcal{C})$ to a statistical model as was described in Section 2. Taking $\varpi(\theta; \lambda) \propto \exp(-\lambda\|\theta\|^2)$ as a prior distribution of θ, the posterior is proportional to $\varpi(\theta; \lambda)\mu(y \mid x; \theta)$ for $\mu \in \mathcal{M}(\mathcal{C})$. It is straightforward to find that the search of the minimizer of the loss function corresponding to $\widehat{\mathbb{A}}$ is equal to the search of the mode of the posterior.

We evaluate the generalization performance of the regularized boosting. An upperbound of the generalization error of kernel machines was derived in Bartlett and Mendelson (2002). Applying their way to this method, we easily obtain the following upperbound of the generalization error independent of the choice of the parameter distribution π in Eq. (7) or (12).

Proposition 2. *Let L be a Lipschitz constant of* exp *in the range of $\overline{F}(x;\eta)$. Let further $K_{\mathcal{C}}$ be any WL kernel function associated with $\mathcal{C}$. With probability at least $1-\delta$, every function $\overline{F}(x;\eta) = \sum_{\ell=1}^{n} \eta_\ell K_{\mathcal{C}}(x, X_\ell)$ with $\|\overline{F}\|^2_{\mathcal{H}_{K_{\mathcal{C}}}} \leq B$ satisfies*

$$P(Y\overline{F}(X;\eta) \leq 0) \leq \widehat{A}(\overline{F}(\cdot,\eta)) + \frac{4LB}{\sqrt{n}} + L\sqrt{\frac{2\ln(1/\delta)}{n}}.$$

This upperbound indicates that the generalization error of $\overline{F}$ is small if $\widehat{A}(\overline{F})$ and B are small, which is just what the proposed algorithm tries to do.

4 Experiments

To illustrate the performance of our proposals, we make a comparison between AdaBoost with $\mathcal{C}_{\rm ds}$ and the proposed regularized AdaBoost (RegAdaBoost) with kernel functions $K_{\mathcal{C}_{\rm ds}}$, $K_{\rm ds}$ and K_g, where K_g is an RBF kernel $K_g(x,x') = \exp(-\|x-x'\|^2/h)$. In experiments, we used the range of each feature in the training data as the estimates of $\{\ell_m, r_m\}$ in $K_{\rm ds}$. The iteration number τ in AdaBoost and λ in RegAdaBoost were determined by using ten-fold cross-validation. The iteration number τ in RegAdaBoost was taken as sufficiently long except the case of K_g. In the case of K_g, we used ten-fold cross validation to choose τ since the use of K_g causes sometimes the overfit. To evaluate the generalization performance, we first partition the whole data set into the training data and the test data. The size of test data was taken as much as possible unless it is larger than a thousand. Each method constructed a classifier based on the training data and its performance (test error) was evaluated on the test data.

The data sets shown in Table 1 were from UCI and DELVE benchmark repositories except 'slope data'. Some of the data sets are not binary classification problems, therefore, we partition the labels into two groups. 'slope data' is the simplest artificial data illustrated in Figure 2 and reflects the characteristics of each method. AdaBoost overfitted to the mislabels, while RegAdaBoost with $K_{\mathcal{C}_{\rm ds}}$ constructs a smooth decision boundary by ignoring them. Table 1 in fact indicates that RegAdaBoost overperformed AdaBoost in almost data sets and gave the best or nearly-best results.

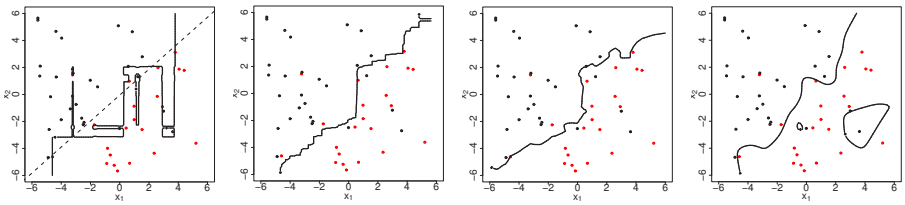
It is observed in Table 1 that RegAdaBoost with $K_{\mathcal{C}_{\rm ds}}$ and $K_{\rm ds}$ exhibited quite similar performance. The similarity of the Gram matrix between them in Figure 1 also supports this observation since RegAdaBoost depends on the Gram matrix directly. These observations implies the superiority of $K_{\rm ds}$ because of its small computational cost.

Finally, it is remarked that RegAdaBoost with RBF kernel is more flexible as illustrated in Figure 2 and 1. As a result, it attains the good approximation ability but suffers from the overfit. Our experiments showed that it performed well in several data but quite poor sometimes. The main reason of the poor performance is that K_g has only one parameter h because we did not use of

multiscaling. However, the tuning of the multiscaling of RBF kernel is not easy. In contrast, the decision stump kernel K_{ds} has parameters that can quite easily be tuned.

Table 1: Performance of AdaBoost and RegAdaBoost with various kernel functions on some data sets. 'sample' is the number of samples, 'feature' is the number of features. 'training data' is the number of the training data. 'TE' denotes the test error. (best method in bold face). 'bcw' is 'breast-cancer-wisconsin' data.

data	type	feature	training data	TE (AdaBoost)	TE (RegAdaBoost)		
					$K_{\mathcal{C}_{\mathrm{ds}}}$	K_{ds}	K_g
slope	artificial	2	50	19.5	**14.7**	14.8	18.1
bcw	real	9	200	7.88	3.32	3.73	**2.9**
splice	real	60	200	13.4	12.4	**12.2**	17.3
thyroid	real	5	100	5.26	**0.87**	**0.87**	3.51
titanic	real	3	600	**20.5**	21.7	20.8	21
wine	real	13	100	5.19	**0**	**0**	31.16
waveform	artificial	21	200	11.1	9.8	9.3	**8.7**



(a) AdaBoost (b) $K_{\mathcal{C}_{\mathrm{ds}}}$ (c) K_{ds} (d) K_g

Figure 2: Plots of 'slope data' in Table 1 with decision boundary (the solid line) constructed by classification methods. The red points belong to the positive class, while the black points belong to the negative class. Panel (a): AdaBoost, Panel (b)-(d): RegAdaBoost with various kernels. The dotted line in the top left panel shows the Bayes optimal decision boundary.

5 Discussion

In this paper, we proposed some extensions of boosting based on the connections between boosting and a kernel machine through the WL kernel. We developed a new simple regularized boosting algorithm (RegAdaBoost) whose regularizer is naturally derived from the WL kernel. It has a more direct connection to the

kernel machines than already proposed regularized boosting methods, and our experiments indicated that this regularized AdaBoost overperformed AdaBoost on various real-world data.

One drawback of the WL kernel is the computational cost, but we overcome this problem by proposing a new kernel, which is based on the investigation of the RKHS induced by the weak learners. Although we have only shown such a kernel for decision stumps weak learners (*decision stump kernel*), it works remarkably well with less computation and we believe this opens a new direction for boosting methods. Our future works are to study the properties of such kernel functions, and extend this idea from decision stumps to wider class of weak learners.

References

Bartlett, P. L., Mendelson, S., 2002. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research* 3, 463–482.

Breiman, L., 1998. Arcing classifiers. *Annals of Statistics* 26 (3), 801–849.

Canu, S., Smola, A. J., 2006. Kernel methods and the exponential family. *Neurocomputing* 69, 714–720.

Freund, Y., Schapire, R. E., 1997. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55 (1), 119–139.

Friedman, J. H., Hastie, T., Tibshirani, R., 2000. Additive logistic regression: A statistical view of boosting. Annals of Statistics 28, 337–407.

Lebanon, G., Lafferty, J., 2002. Boosting and maximum likelihood for exponential models. *Advances in Neural Information Processing Systems* 14.

Murata, N., Takenouchi, T., Kanamori, T., Eguchi, S., 2004. Information geometry of U-Boost and Bregman divergence. *Neural Computation* 16, 1437–1481.

Rätsch, G., Onoda, T., Müller, K. R., 1999. Regularizing AdaBoost. *Neural Information Processing System*.

Rätsch, G., Onoda, T., Müller, K. R., 2001. Soft margins for AdaBoost. *Machine Learning* 42, 287–320.

Rätsch, G., Schölkopf, B., Mika, S., Müller, K. R., 2000. SVM and Boosting: One Class. *GMD FIRST* 119.

Schapire, R., Singer, Y., 1999. Improved boosting algorithms using confidence-rated predictions. *Machine Learning* 37 (3), 297–336.

Schölkopf, B., Smola, A. J., 2001. Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond. MIT Press.

Sun, Y., Li, J., Hager, W., 2004. Two new regularized AdaBoost algorithms. *ICMLA*.

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