A Bayesian Framework for Online Classifier Ensemble

Qinxun Bai QINXUN@CS.BU.EDU
Department of Computer Science, Boston University, Boston, MA 02215 USA

Henry Lam KHLAM@BU.EDU
Department of Mathematics and Statistics, Boston University, Boston, MA 02215 USA

Stan Sclaroff SCLAROFF@CS.BU.EDU
Department of Computer Science, Boston University, Boston, MA 02215 USA

Abstract

We propose a Bayesian framework for recursively estimating the classifier weights in online learning of a classifier ensemble. In contrast with past methods, such as stochastic gradient descent or online boosting, our framework estimates the weights in terms of evolving posterior distributions. For a specified class of loss functions, we show that it is possible to formulate a suitably defined likelihood function and hence use the posterior distribution as an approximation to the global empirical loss minimizer. If the stream of training data is sampled from a stationary process, we can also show that our framework admits a superior rate of convergence to the expected loss minimizer than is possible with standard stochastic gradient descent. In experiments with real-world datasets, our formulation often performs better than online boosting algorithms.

1. Introduction

The literature on online ensemble classification has studied recursive mechanisms to combine several weak classifiers, when given labeled training data $\{\mathbf{x}_t, y_t\}_{t=1}^T$ that arrive sequentially. Different approaches have been proposed, including online extensions of boosting (Oza & Russell, 2001; Pelossof et al., 2009) and stochastic gradient descent based methods (Babenko et al., 2009b; Leistner et al., 2009; Grbovic & Vucetic, 2011). Recently, Chen et al. (2012) formulated a smoothed boosting algorithm based on the analysis of regret from offline benchmarks.

In this paper, we pose the online ensemble problem as a loss minimization problem with respect to the ensemble weights, and propose an online ensemble classification method that is not based on boosting or gradient descent. The main idea is to recursively estimate a posterior distribution of the ensemble weights in a Bayesian manner. We show that, for a given class of loss functions, we can define a likelihood function on the ensemble weights and, with an appropriately formulated prior distribution, we can generate a posterior mean that closely approximates the empirical loss minimizer.

Our proposed scheme is straightforward, but powerful in two respects. First, it can approximate the global optimal solution, in contrast with local methods such as stochastic gradient descent (SGD). Second, assuming the training data is sampled from a stationary process, our Bayesian scheme possesses a rate of convergence to the expected loss minimizer that is at least as fast as standard SGD. In fact, our rate is faster unless the SGD step size is chosen optimally, which cannot be done *a priori* in the online setting. We identify the class of loss functions where both of the above properties are precisely satisfied. In experiments with real-world datasets, our formulation often performs better than state-of-the-art online boosting algorithms.

2. Related Work

A number of past works focus on online learning with concept drift (Wang et al., 2003; Kolter & Maloof, 2005; 2007; Minku, 2011), which differs from stationary online settings. Given the technical difficulty, theoretical analysis for concept drift seems to be underdeveloped. Kolter & Maloof (2005) proved error bounds for their proposed method, which appears to be the first such theoretical analysis, yet such analysis is not easily generalized to other methods in this category. Other works, such as Schapire (2001) and Cesa-Bianchi & Lugosi (2003), obtained performance bounds from the perspective of iterative games.

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

Our work is more closely related to methods that operate in a stationary environment, most notably online boosting methods. One of the first methods was proposed by Oza & Russell (2001), who showed asymptotic convergence to batch boosting under certain conditions. However, the convergence result only holds for some simple "lossless" weak learners (Oza, 2001), such as Naive Bayes. Other variants of online boosting have been proposed, such as methods that employ feature selection (Grabner & Bischof, 2006; Liu & Yu, 2007), semi-supervised learning (Grabner et al., 2008), multiple instance learning (Babenko et al., 2009a), and multi-class learning (Saffari et al., 2010). However, most of these works consider the design and update of weak learners beyond that of (Oza, 2001) and, thus, do not bear the convergence guarantee therein. Other methods employ the gradient descent framework, such as Online GradientBoost (Leistner et al., 2009), Online Stochastic Boosting (Babenko et al., 2009b) and Incremental Boosting (Grbovic & Vucetic, 2011). Many of these methods possess convergence results, which provide a basis for comparison with our framework. In fact, we show that our method compares favorably to gradient descent in terms of asymptotic convergence rate. Lastly, Chen et al. (2012) proposed an online boosting method with a theoretical bound on the error rate, with the novel design of a smoother and more conservative update of the online weak classifiers.

Our idea is related to, yet differs from, simulated annealing (Laarhoven et al., 1987) and Bayesian model averaging (Hoeting et al., 1999). The former is a global optimization technique, typically conducted by defining a probability distribution that has the objective function that one wants to minimize as an exponent, and running Monte Carlo to estimate the peak of this distribution. Simulated annealing, nevertheless, is primarily motivated for deterministic global optimization, and should be contrasted with the stochastic and also the sequential nature of our framework. Next, conventional Bayesian model averaging aims to combine several plausible models as a closer description of the data. In contrast, our Bayesian framework does not focus on the actual model that generates the data, but is instead motivated as a loss minimization algorithm.

3. Bayesian Recursive Ensemble

We denote the input feature by $\mathbf{x}$ and its classification label by y (1 or -1). We assume that we are given m binary weak classifiers $\{c_i(\mathbf{x})\}_{i=1}^m$, and our goal is to find the best ensemble weights $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_m)$ where $\lambda_i \geq 0$, to construct an ensemble classifier. For now, we do not impose a particular form of ensemble method (we defer this until Section 4), although one example form is $\sum_i \lambda_i c_i(\mathbf{x})$. We focus on online learning, where training data $(\mathbf{x}, y)$ comes in sequentially, one at a time at $t = 1, 2, 3, \ldots$.

3.1. Loss Specification

We first introduce a loss function at the weak classifier level. Given a training pair $(\mathbf{x}, y)$ and an arbitrary weak classifier h, we denote $g := g(h(\mathbf{x}), y)$ as a non-negative loss function. Possible choices of g include the logistic loss function, hinge loss, zero-one loss, etc. If h is one of the given weak classifiers c_i, we will denote $g(c_i(\mathbf{x}), y)$ as $g_i(\mathbf{x}, y)$, or simply g_i for ease of notation. Furthermore, we define $g_i^t := g(c_i^t(\mathbf{x}^t), y^t)$ where $(\mathbf{x}^t, y^t)$ is the training sample and c_i^t the updated i-th weak classifier at time t. To simplify notation, we use $\mathbf{g} := (g_1, \ldots, g_m)$ to denote the vector of losses for the weak classifiers, $\mathbf{g}^t := (g_1^t, \ldots, g_m^t)$ to denote the losses at time t, and $\mathbf{g}^{1:T} := (\mathbf{g}^1, \ldots, \mathbf{g}^T)$ to denote the losses up to time T.

With the above notation, we let $\ell_t(\boldsymbol{\lambda}; \mathbf{g}^t)$ be some ensemble loss function at time t, which depends on the ensemble weights and the individual loss of each weak classifier. We then define our cumulative ensemble loss as follows:

$$L_T(\boldsymbol{\lambda}; \mathbf{g}^{1:T}) = \ell_0(\boldsymbol{\lambda}) + \sum_{t=1}^{T} \ell_t(\boldsymbol{\lambda}; \mathbf{g}^t) \qquad (1)$$

where $\ell_0(\boldsymbol{\lambda})$ can be regarded as an initial loss, which becomes negligible as T progresses.

We make two sets of assumptions on L_T that are adapted from Chen (1985): one on the regularity conditions on L_T, the other on the form of ℓ_t to ensure eligibility in applying our Bayesian approach. We now specify these assumptions.

Assumption 1 (Regularity conditions)**.** *Assume that for each T, there exists a $\boldsymbol{\lambda}_T^*$ that minimizes* (1)*, and*

1. *"local optimality": for each T, $\nabla L_T(\boldsymbol{\lambda}_T^*; \mathbf{g}^{1:T}) = 0$ and $\nabla^2 L_T(\boldsymbol{\lambda}_T^*; \mathbf{g}^{1:T})$ is positive definite.*
2. *"steepness": the minimum eigenvalue of $\nabla^2 L_T(\boldsymbol{\lambda}_T^*; \mathbf{g}^{1:T})$ diverges to ∞ as $T \to \infty$.*
3. *"smoothness": For any $\epsilon > 0$, there exists a positive integer N and $\delta > 0$ such that for any $T > N$ and $\boldsymbol{\theta} \in H_\delta(\boldsymbol{\lambda}_T^*) = \{\boldsymbol{\theta} : \|\boldsymbol{\theta} - \boldsymbol{\lambda}_T^*\|_2 \leq \delta\}$, $\nabla^2 L_T(\boldsymbol{\theta}; \mathbf{g}^{1:T})$ exists and satisfies*

$$I - A(\epsilon) \leq \nabla^2 L_T(\boldsymbol{\theta}; \mathbf{g}^{1:T}) \left(\nabla^2 L_T(\boldsymbol{\lambda}_T^*; \mathbf{g}^{1:T})\right)^{-1} \leq I + A(\epsilon)$$

   for some positive semidefinite symmetric matrix A whose largest eigenvalue tends to 0 as $\epsilon \to 0$, and the inequalities above are matrix inequalities.
4. *"concentration": for any $\delta > 0$, there exists a positive integer N and constants $c, p > 0$ such that for any $T > N$ and $\theta \notin H_\delta(\boldsymbol{\lambda}_T^*)$, we have*

$$L_T(\boldsymbol{\theta}; \mathbf{g}^{1:T}) - L_T(\boldsymbol{\lambda}_T^*; \mathbf{g}^{1:T}) < c\left((\boldsymbol{\theta} - \boldsymbol{\lambda}_T^*)' \nabla^2 L_T(\boldsymbol{\lambda}_T^*; \mathbf{g}^{1:T})(\boldsymbol{\theta} - \boldsymbol{\lambda}_T^*)\right)^p$$

In the situation where ℓ_t is separable in terms of each component of $\boldsymbol{\lambda}$, i.e. $\ell_t(\boldsymbol{\lambda};\mathbf{g}) = \sum_{i=1}^m r_i(\lambda_i;\mathbf{g})$ and $\ell_0(\boldsymbol{\lambda}) = \sum_{i=1}^m s_i(\lambda_i)$ for some twice differentiable functions $r_i(\cdot;\mathbf{g})$ and $s_i(\cdot)$, the assumptions above will depend only on $f_i(\lambda;\mathbf{g}^{1:T}) := \sum_{t=1}^T r_i(\lambda;\mathbf{g}^t) + s_i(\lambda)$ for each i. For example, Condition 3 in Assumption 1 reduces to merely checking uniform continuity of each $f_i''(\cdot;\mathbf{g}^{1:T})$.

Condition 1 in Assumption 1 can be interpreted as the standard first and second order conditions for the optimality of $\boldsymbol{\lambda}_T^*$, whereas Condition 3 in essence requires continuity of the Hessian matrix. Conditions 2 and 4 are needed for the use of Laplace method (Cox & Hinkley, 1974), which, as we will show later, stipulates that the posterior distribution peaks near the optimal solution $\boldsymbol{\lambda}_T^*$.

Assumption 2 (Density interpretation)**.** *The loss functions ℓ_t satisfy*

$$\int e^{-\ell_t(\boldsymbol{\lambda};\mathbf{z})} d\mathbf{z} = 1 \tag{2}$$

for $t = 1, 2, \ldots$, and ℓ_0 satisfies

$$\int e^{-\ell_0(\mathbf{w})} d\mathbf{w} = 1. \tag{3}$$

In view of (1), ℓ_0 does not contribute significantly to the cumulative loss as T increases, and it can be specified by the user on the basis of convenience. The condition in (2) is more crucial, and requires that the exponent of the loss function $\ell_t(\boldsymbol{\lambda};\cdot)$ behaves exactly as a probability density.

3.2. A Bayesian Framework

Loss functions ℓ_t that satisfy Assumptions 1 and 2 can be used to define $p_t(\mathbf{g}|\boldsymbol{\lambda}) = e^{-\ell_t(\boldsymbol{\lambda};\mathbf{g})}$ as a likelihood function for $\mathbf{g}$, parametrized by $\boldsymbol{\lambda}$ and $p_0(\boldsymbol{\lambda}) = e^{-\ell_0(\boldsymbol{\lambda})}$, as a prior for the parameter $\boldsymbol{\lambda}$. Our update scheme for $\boldsymbol{\lambda}$ then hinges on calculating the posterior mean for $\boldsymbol{\lambda}$ at each step. A summary of this algorithm is given in Algorithm 1.

Algorithm 1 Bayesian Ensemble

Input: streaming samples $\{(\mathbf{x}^t, y^t)\}_{t=1}^T$
online weak learners $\{c_i^t(\mathbf{x})\}_{i=1}^m$
chosen likelihood $p(\mathbf{g}|\boldsymbol{\lambda})$ and prior $p(\boldsymbol{\lambda})$
Initialize: hyper-parameters for $p(\mathbf{g}|\boldsymbol{\lambda})$ and $p(\boldsymbol{\lambda})$
for $t = 1$ **to** T **do**
$\forall i$, compute $g_i^t = g(c_i^t(\mathbf{x}^t), y^t)$
update for the posterior distribution of $\boldsymbol{\lambda}$:
$p(\boldsymbol{\lambda}|\mathbf{g}^{1:t}) \propto p(\mathbf{g}^t|\boldsymbol{\lambda}) p(\boldsymbol{\lambda}|\mathbf{g}^{1:t-1}) \propto \prod_{s=1}^t p(\mathbf{g}^s|\boldsymbol{\lambda}) p(\boldsymbol{\lambda})$
update the weak learners using $(\mathbf{x}^t, y^t)$
end for

Algorithm 1 offers the following desirable property.

Theorem 1. *Under Assumptions 1 and 2, the Bayesian scheme in Algorithm 1 produces a posterior distribution $p_T(\boldsymbol{\lambda}|\mathbf{g}^{1:T})$ satisfying the asymptotic normality property*

$$\left(\nabla^2 L_T(\boldsymbol{\lambda}_T^*;\mathbf{g}^{1:T})\right)^{1/2} (\boldsymbol{\lambda}_T - \boldsymbol{\lambda}_T^*) \xrightarrow{d} N(0,1) \tag{4}$$

where $\boldsymbol{\lambda}_T$ is interpreted as a random variable with distribution $p_T(\boldsymbol{\lambda}|\mathbf{g}^{1:T})$, and $\xrightarrow{d}$ denotes convergence in distribution. Furthermore, under the uniform integrability condition $\sup_T E_{\boldsymbol{\lambda}_T|\mathbf{g}^{1:T}} \|\boldsymbol{\lambda}_T - \boldsymbol{\lambda}_T^*\|_1^{1+\epsilon} < \infty$ for some $\epsilon > 0$, we have

$$|E_{\boldsymbol{\lambda}_T|\mathbf{g}^{1:T}}[\boldsymbol{\lambda}_T] - \boldsymbol{\lambda}_T^*| = o\left(\frac{1}{\sigma_T^{1/2}}\right) \tag{5}$$

where $E_{\boldsymbol{\lambda}_T|\mathbf{g}^{1:T}}[\cdot]$ denotes the posterior expectation and σ_T is the minimum eigenvalue of the matrix $\nabla^2 L_T(\boldsymbol{\lambda}_T^*;\mathbf{g}^{1:T})$.

The idea behind (4) comes from a classical technique in Bayesian asymptotics known as the Laplace method (Cox & Hinkley, 1974). Theorem 1 states that given the loss structure satisfying Assumptions 1 and 2, the posterior distribution of $\boldsymbol{\lambda}$ under our Bayesian update scheme provides an approximation to the minimizer $\boldsymbol{\lambda}_T^*$ of the cumulative loss at time T, as T increases, by tending to a normal distribution peaked at $\boldsymbol{\lambda}_T^*$ with shrinking variance. The bound (5) states that this posterior distribution can be summarized using the posterior mean to give a point estimate of $\boldsymbol{\lambda}_T^*$. Moreover, note that $\boldsymbol{\lambda}_T^*$ is the global, not merely local, minimizer of the cumulative loss. This approximation of global optimum highlights a key advantage of the Bayesian scheme over other methods such as stochastic gradient descent (SGD), which only find a local optimum.

The next theorem states another benefit of our Bayesian scheme over standard SGD. Supposing that SGD does indeed converge to the global optimum. Even so, it turns out that the Bayesian scheme converges faster than standard SGD under the assumption of i.i.d. training samples.

Theorem 2. *Suppose Assumptions 1 and 2 hold. Assume also that $\ell_t(\boldsymbol{\lambda};\mathbf{g}) = \ell(\boldsymbol{\lambda};\mathbf{g})$ are identical across t and $\mathbf{g}^t$ are i.i.d., with $E[\ell(\boldsymbol{\lambda};\mathbf{g})] < \infty$ and $E[\ell(\boldsymbol{\lambda};\mathbf{g})^2] < \infty$. The Bayesian posterior mean produced by Alg. 1 converges to $\operatorname{argmin}_{\boldsymbol{\lambda}} E[\ell(\boldsymbol{\lambda};\mathbf{g})]$ strictly faster than standard SGD (supposing it converges to the global minimum), given by*

$$\boldsymbol{\lambda}_{T+1} \leftarrow \boldsymbol{\lambda}_T - \epsilon_T K \nabla \ell(\boldsymbol{\lambda}_T;\mathbf{g}^T) \tag{6}$$

in terms of the asymptotic variance, except when the step size ϵ_T and the matrix K is chosen optimally.

In Theorem 2, by asymptotic variance we mean the following: it turns out that both the posterior mean and the update from SGD possess versions of the central limit theorem, in the form $\sqrt{T}(\boldsymbol{\lambda}_T - \boldsymbol{\lambda}^*) \xrightarrow{d} N(0,\Sigma)$ where $\boldsymbol{\lambda}^* =$

$\text{argmin}_{\boldsymbol{\lambda}} E[\ell(\boldsymbol{\lambda}; \mathbf{g})]$. Our comparison is on the asymptotic variances Σ: the smaller Σ is, the smaller the multiplicative constant in the rate of convergence $1/\sqrt{T}$.

The result follows from first comparing central limit theorems for the minimizer of cumulative loss $\boldsymbol{\lambda}_T^*$ and the standard SGD update in (6), and secondly, from the relation (5), which states that the difference between the posterior mean and $\boldsymbol{\lambda}_T^*$ decreases at a rate faster than the central limit theorem, and hence is asymptotically negligible.

4. Loss Functions, Likelihoods and Priors

We first discuss in depth a simple and natural choice of loss function and its corresponding likelihood function and prior, which are also used in our experiments in Section 5. Then we briefly give examples of other loss functions that also fit into our framework.

4.1. Exponential Likelihood and Gamma Prior

For any t, consider

$$\ell_t(\boldsymbol{\lambda}; \mathbf{g}) = \theta \sum_{i=1}^m \lambda_i g_i - \sum_{i=1}^m \log \lambda_i \tag{7}$$

The motivation for (7) is straightforward: it is the sum of individual loss each weighted by λ_i. The extra term $\log \lambda_i$ prevents λ_i from approaching zero, the trivial minimizer for the first term. The parameter θ specifies the trade-off between the importance of the first and the second term. This loss function satisfies Assumptions 1 and 2. In particular, the Hessian of L_T turns out to not depend on $g^{1:T}$, therefore all conditions of Assumption 1 can be verified easily. For Assumption 2, the exponent of the negation of the loss function in (7), plus a constant term $m \log \theta$, which does not affect the loss minimization, integrates to 1.

Using the discussion in Section 3.2, we choose the exponential likelihood

$$p(\mathbf{g}|\boldsymbol{\lambda}) = \prod_{i=1}^m (\theta \lambda_i) e^{-\theta \lambda_i g_i} \tag{8}$$

To facilitate computation, we employ the Gamma prior:

$$p(\boldsymbol{\lambda}) \propto \prod_{i=1}^m \lambda_i^{\alpha-1} e^{-\beta \lambda_i} \tag{9}$$

where α and β are the hyper shape and rate parameters. Correspondingly, we pick $\ell_0(\boldsymbol{\lambda}) = \beta \sum_{i=1}^m \lambda_i - (\alpha - 1) \sum_{i=1}^m \log \lambda_i$. To be concrete, the cumulative loss in (1) (disregarding the constant terms) is

$$\beta \sum_{i=1}^m \lambda_i - (\alpha-1) \sum_{i=1}^m \log \lambda_i + \sum_{t=1}^T \left(\theta \sum_{i=1}^m \lambda_i g_i^t - \sum_{i=1}^m \log \lambda_i \right)$$

Now, under conjugacy of (8) and (9), the posterior distribution of $\boldsymbol{\lambda}$ after t steps is given by the Gamma distribution

$$p(\boldsymbol{\lambda}|\mathbf{g}^{1:t}) \propto \prod_{i=1}^m (\lambda_i)^{\alpha+t-1} e^{-(\beta + \theta \sum_{s=1}^t g_i^s)\lambda_i}$$

Therefore the posterior mean for each λ_i is

$$\frac{\alpha + t}{\beta + \theta \sum_{s=1}^t g_i^s} \tag{10}$$

We use the following prediction rule at each step:

$$y = \begin{cases} 1 & \text{if } \sum_{i=1}^m \lambda_i g_i(\mathbf{x}, 1) \le \sum_{i=1}^m \lambda_i g_i(\mathbf{x}, -1) \\ -1 & \text{otherwise} \end{cases} \tag{11}$$

where each λ_i is the posterior mean given by (10). For this setup, Algorithm 1 can be cast as Algorithm 2 below, which is to be implemented in the numerical section.

Algorithm 2 Closed-form Bayesian Ensemble

Input: streaming samples $\{(\mathbf{x}^t, y^t)\}_{t=1}^T$
online weak learners $\{c_i^t(\mathbf{x})\}_{i=1}^m$
Initialize: parameters θ for likelihood (8) and parameters α, β for prior (9)
for $t = 1$ **to** T **do**
$\forall i$, compute $g_i^t = g(c_i^t(\mathbf{x}^t), y^t)$, where g is logistic loss function
update the posterior mean of $\boldsymbol{\lambda}$ by (10)
update the weak learners according to the particular choice of online weak learner
make prediction by (11) for next incoming sample
end for

The following bound provides further understanding of the loss function (7) and the prediction rule (11), by relating their use with a guarantee on the prediction error:

Theorem 3. *Suppose that $\mathbf{g}^t$ are i.i.d., so that $\boldsymbol{\lambda}_T^*$ converges to $\boldsymbol{\lambda}^* := \text{argmin}_{\boldsymbol{\lambda}} E[\ell(\boldsymbol{\lambda}; \mathbf{g})]$ for ℓ defined in (7). The prediction error using rule (11) with $\boldsymbol{\lambda}^*$ is bounded by*

$$P_{(\mathbf{x},y)}(\textit{error}) \le m^{\frac{1}{p}} \left(E_{(\mathbf{x},y)} \left[\left(\sum_{i=1}^m \frac{g_i(\mathbf{x}, -y)}{E[g_i(\mathbf{x}, y)]} \right)^{\frac{-1}{p-1}} \right] \right)^{\frac{p-1}{p}} \tag{12}$$

for any $p > 1$.

The proof, given in the Appendix, provides a bound for the long-run prediction error under stationary environment. To make sense of this result, note that the quantity $\frac{1}{E[g_i(\mathbf{x},y)]} g_i(\mathbf{x}, -y)$ can be interpreted as a performance indicator of each weak classifier, i.e. the larger it is, the better the weaker classifier is, since a good classifier should

have a small loss $E[g_i(\mathbf{x}, y)]$ and correspondingly a large $g_i(\mathbf{x}, -y)$. As long as there exists some good weak classifiers among the m choices, $\sum_{i=1}^{m} \frac{g_i(\mathbf{x},-y)}{E[g_i(\mathbf{x},y)]}$ will be large, which leads to a small error bound in (12).

Finally, in correspondence to Theorem 2, the SGD for (7) is written as

$$\lambda_i^{t+1} = \lambda_i^t - \frac{\gamma}{t}\left(\theta g_i^t - \frac{1}{\lambda_i^t}\right) \tag{13}$$

where γ is a parameter that controls the step size. The following result is a consequence of Theorem 2 (a proof for this particular case appears in the Appendix).

Theorem 4. *Suppose that* $\mathbf{g}^t$ *are i.i.d., and* $0 < E_{(\mathbf{x},y)}[g_i(\mathbf{x}, y)] < \infty$ *and* $Var_{(\mathbf{x},y)}(g_i(\mathbf{x}, y)) < \infty$. *For each* λ_i, *the posterior mean given by* (10) always *has a rate of convergence at least as fast as the SGD update* (13) *in terms of asymptotic variance. In fact, it is strictly better in all situations except when the step size parameter* γ *in* (13) *is set optimally* a priori.

In Section 5 we will see that this simple choice of loss function and Bayesian update scheme lead to superior empirical performance compared with other methods.

4.2. Other Examples

Note that while our simple choice of (7) can be directly minimized, this is not true in general. We now give a few other examples of acceptable ensemble loss functions.

1. The loss function

$$\begin{aligned}\ell_t(\boldsymbol{\lambda}; \mathbf{g}) &= \sum_{i=1}^{m}(1-\lambda_i)\log g_i + \theta\sum_{i=1}^{m} g_i \\ &+ \sum_{i=1}^{m}\log\Gamma(\lambda_i) - (\log\theta)\sum_{i=1}^{m}\lambda_i\end{aligned}$$

   where $\theta > 0$ is a parameter, corresponds to the product of Gamma likelihood given by

$$p(\mathbf{g}|\boldsymbol{\lambda}) = \prod_{i=1}^{m}\frac{\theta^{\lambda_i}}{\Gamma(\lambda_i)} g_i^{\lambda_i - 1} e^{-\theta g_i}$$

   A conjugate prior for $\boldsymbol{\lambda}$ is available, in the form

$$p(\boldsymbol{\lambda}) \sim \prod_{i=1}^{m}\frac{a^{\lambda_i - 1}\theta^{c\lambda_i}}{\Gamma(\lambda_i)^b}$$

   where $a, b, c > 0$ are hyper parameters.

2. We can generalize the ensemble weights to include two parameters $\lambda_i = (\alpha_i, \beta_i)$. In this case, we may define the loss function as

$$\begin{aligned}\ell_t(\boldsymbol{\alpha}, \boldsymbol{\beta}; \mathbf{g}) &= \sum_{i=1}^{m}\beta_i g_i + \sum_{i=1}^{m}(1-\alpha_i)\log g_i \\ &+ \sum_{i=1}^{m}\log\Gamma(\alpha_i) - \sum_{i=1}^{m}\alpha_i\log\beta_i\end{aligned}$$

   with the following Gamma likelihood

$$p(\mathbf{g}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \prod_{i=1}^{m}\frac{\beta_i^{\alpha_i}}{\Gamma(\alpha_i)} g^{\alpha_i - 1} e^{-\beta_i g_i}$$

   A conjugate prior is available for $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ jointly

$$p(\boldsymbol{\alpha}, \boldsymbol{\beta}) \sim \prod_{i=1}^{m}\frac{p^{\alpha_i - 1} e^{-\beta_i q}}{\Gamma(\alpha_i)^r \beta_i^{-\alpha_i s}}$$

   where p, q, r, s are hyper parameters.

5. Experiments

We report two sets of experiments on binary classification benchmark datasets[1]. In the first set of experiments, we evaulate our scheme's performance vs. three baseline methods, given static pre-trained weak learners. In the second set of experiments, we compare with leading online bosting methods, following the setup of (Chen et al., 2012).

Following (Chen et al., 2012), we report experiments using two different weak learners: Perceptron and Naive Bayes. In every trial, each ensemble method is given 100 weak learners and the average error rate is reported over 5 random trials of different orders of each dataset.

In all experiments, we have set the hyperparameters of our method $\alpha = \beta = 1$ and $\theta = 0.1$. From the expression of posterior mean (10), the prediction rule (11) is unrelated to the values of α, β and θ in long term. In experiments, we also find that our method achieves stable results with respect to settings of these parameters. However, the stochastic gradient descent baseline (SGD) (13) is sensitive to θ; therefore, use the same value $\theta = 0.1$ in our method.

5.1. Comparison with Baseline Methods

We compare our online ensemble method with three baseline methods, using two different weak learners: PERCEPTRON and NAIVE BAYES. The first baseline is a single Perceptron/Naive Bayes classifier. The second baseline (VOTING) is the uniform ensemble of weak learners. The third baseline (SGD) is the ensemble method with ensemble weights estimated by stochastic gradient descent (13). OURS is our proposed Bayesian ensemble method. The same static weak learners are shared by all ensemble methods. Each data set is split into training and testing sets for each random trial, where a training set contains no more than 10% of the total amount of data. In order to make weak learners divergent, a weak learner uses a randomly

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

Table 1. Error rate for our method vs baselines, using online Perceptron/Naive Bayes weak learners.

		PERCEPTRON WEAK LEARNER				NAÏVE BAYES WEAK LEARNER			
DATA SET	# EXAMPLES	PERCEPTRON	VOTING	SGD	OURS	NAÏVE BAYES	VOTING	SGD	OURS
HEART	270	0.287	0.275	0.261	**0.249**	0.249	0.214	0.216	**0.210**
BREAST-CANCER	683	0.062	0.052	0.051	**0.041**	0.048	0.044	0.043	**0.039**
AUSTRALIAN	693	0.212	0.209	0.167	**0.143**	0.248	0.241	0.240	**0.225**
DIABETES	768	0.356	0.339	0.321	**0.293**	0.299	0.284	0.283	**0.278**
GERMAN	1000	0.359	0.337	0.334	**0.323**	0.348	0.327	0.315	**0.312**
SPLICE	3175	0.393	0.371	0.331	**0.317**	0.169	0.174	0.169	**0.160**
MUSHROOMS	8124	0.120	0.104	0.102	**0.078**	0.032	0.049	0.047	**0.026**
IONOSPHERE	351	0.279	0.260	0.257	**0.249**	0.199	0.207	0.201	**0.190**
SONAR	208	0.414	0.400	0.392	**0.388**	0.306	0.305	0.304	**0.300**
SVMGUIDE3	1284	0.385	0.437	0.414	**0.332**	0.311	0.299	0.281	**0.240**

sampled subset of data features as input for both training and testing. The first baseline always uses all the features.

Classifier error rates for this experiment are shown in Table 1. Our proposed method consistently performs the best for all datasets. Its superior performance against the voting baseline is consistent with the asymptotic convergence result given by Theorem 1. Its superior performance against the SGD baseline is consistent with the convergence rate analysis given by Theorem 4. To better see this convergence rate difference, we produced plots of the error rate between our method and the SGD baseline, as online learning progresses (see supplemental material).

5.2. Comparison with Online Boosting Methods

We further compare our method with three representative online boosting methods: OZABOOST (Oza & Russell, 2001), OGBOOST is the online GradientBoost method proposed by (Leistner et al., 2009), and OSBOOST is the online Smooth-Boost proposed by Chen et al. (2012). Our method is trained and compared following their setup. However, we discard the "Ijcnn1" and "Web Page" datasets from the tables of (Chen et al., 2012), because they are highly biased with portions of positive samples around 0.09 and 0.03 respectively, and even a naive "always negative" classifier achieves a comparable performance.

The error rates for this experiment are shown in Tables 2 and 3. Our method consistently outperforms competing methods for the Perceptron weak learner and performs among the best for the Naive Bayes weak learner. It is worth noting that our method is the only one that outperforms the baseline in all data sets, which further confirms the effectiveness of the proposed ensemble scheme.

We also note that despite our best efforts to align both the weak learner construction and experiment setup with competing methods (Chen et al., 2012; Chen, 2013), there are inevitably differences in weak learner construction. Firstly, given that our method only focuses on optimizing the ensemble weights, each incoming sample is treated equally in the update of all weak learners, while all three online boosting methods adopt more sophisticated weighted update of weak learners, where the sample weight is dynamically adjusted during each round of update. Secondly, in order to make weak learners different from each other, our weak learners use only a subset of input features, while weak learners of competing methods use all features and update them differently. As a result, the weak learners used by our method are actually weaker than in competing methods. Nevertheless, our method often compares favorably.

6. Future Work

This work can be viewed as an initial attempt to explore the proposed Bayesian framework in the context of online learning of classifier ensembles. Potential followup works include the analysis of sequential Monte Carlo (Doucet & Johansen, 2009) for non-closed-form Bayesian update, the extension of our analysis to non-stationary environments, the application of our framework to more general ensemble rules, and comparison studies between our method and some more sophisticated SGD-based schemes, such as Polyak-Ruppert averaging.

Acknowledgment: This work was supported in part by US NSF Grants 0910908 and 0965579.

A. Appendix

A.1. Proof of Theorem 1

Proof. The convergence in (4) follows from Theorem 2.1 in (Chen, 1985). The first condition in Assumption 1 is equivalent to conditions (P1) and (P2) therein, while the second and third conditions correspond to (C1) and (C2). The last condition is equivalent to (C3.1), which then implies (C3) there to invoke its Theorem 2.1 to conclude (4).

Table 2. Error rate using online Perceptron weak learner, for our method vs methods as reported in (Chen et al., 2012).

DATA SET	# EXAMPLES	PERCEPTRON	OZABOOST	OGBOOST	OSBOOST	OURS
HEART	270	0.2489	0.2356	0.2267	0.2356	**0.2134**
BREAST-CANCER	683	0.0592	0.0501	0.0445	0.0466	**0.0419**
AUSTRALIAN	693	0.2099	0.2012	0.1962	0.1872	**0.1655**
DIABETES	768	0.3216	0.3169	0.3313	0.3185	**0.3098**
GERMAN	1000	0.3256	0.3364	0.3142	0.3148	**0.3105**
SPLICE	3175	0.2717	0.2759	0.2625	0.2605	**0.2584**
MUSHROOMS	8124	0.0148	0.0080	0.0068	**0.0060**	0.0062
ADULT	48842	0.2093	0.2045	0.2080	0.1994	**0.1682**
COD-RNA	488565	0.2096	0.2170	0.2241	0.2075	**0.1934**
COVERTYPE	581012	0.3437	0.3449	0.3482	0.3334	**0.3115**

Table 3. Error rate using online Naive Bayes weak learner, for our method vs methods as reported in (Chen et al., 2012). For "Cod-RNA" our implementation of the Naive Bayes baseline was unable to duplicate the reported result; ours gave 0.2555 instead.

DATA SET	# EXAMPLES	NAIVE BAYES	OZABOOST	OGBOOST	OSBOOST	OURS
HEART	270	0.1904	0.2570	0.3037	0.2059	**0.1755**
BREAST-CANCER	683	0.0474	0.0635	0.1004	0.0489	**0.0408**
AUSTRALIAN	693	0.1751	0.2133	0.2826	0.1849	**0.1611**
DIABETES	768	0.2664	0.3091	0.3292	0.2622	**0.2467**
GERMAN	1000	0.2988	0.3206	0.3598	0.2730	**0.2667**
SPLICE	3175	0.2520	0.1563	0.1863	0.1370	**0.1344**
MUSHROOMS	8124	0.0076	0.0049	0.0229	**0.0029**	0.0054
ADULT	48842	0.2001	0.1912	0.1878	**0.1581**	0.1658
COD-RNA	488565	0.2206∗	0.0796	0.0568	**0.0581**	0.2552
COVERTYPE	581012	0.3518	0.3293	0.3732	0.3634	**0.3269**

To show the bound (5) we take expectation on (4) to get

$$\left(\nabla^2 L_T(\boldsymbol{\lambda}_T^*; \mathbf{g}^{1:T})\right)^{\frac{1}{2}} \left(E_{\boldsymbol{\lambda}_T|\mathbf{g}^{1:T}}[\boldsymbol{\lambda}_T] - \boldsymbol{\lambda}_T^*\right) \to 0 \quad (14)$$

which is valid because of the uniform integrability condition $\sup_T E_{\boldsymbol{\lambda}_T|\mathbf{g}^{1:T}} \|\boldsymbol{\lambda}_T - \boldsymbol{\lambda}_T^*\|_1^{1+\epsilon} < \infty$ (Durrett, 2010). Therefore, $E_{\boldsymbol{\lambda}_T|\mathbf{g}^{1:T}}[\boldsymbol{\lambda}_T] - \boldsymbol{\lambda}_T^* = \left(\nabla^2 L(\boldsymbol{\lambda}_T^*; \mathbf{g}^{1:T})\right)^{-\frac{1}{2}} \mathbf{w}_T$ where $\mathbf{w}_T = o(1)$ by (14). But then

$$\begin{aligned} & \left\| \left(\nabla^2 L_T(\boldsymbol{\lambda}_T^*; \mathbf{g}^{1:T})\right)^{-\frac{1}{2}} \mathbf{w}_T \right\|_1 \\ \leq & \left\| \left(\nabla^2 L_T(\boldsymbol{\lambda}_T^*; \mathbf{g}^{1:T})\right)^{-\frac{1}{2}} \right\|_1 \|\mathbf{w}_T\|_1 \\ \leq & \frac{C}{\sigma_T^{1/2}} \|\mathbf{w}_T\|_1 = o\left(\frac{1}{\sigma_T^{1/2}}\right) \end{aligned}$$

where $\|\cdot\|_1$ when applied to matrix is the induced L_1-norm. This shows (5). □

A.2. Proof of Theorem 2

Proof. The proof follows by combining (5) with established central limit theorems for sample average approximation (Pasupathy & Kim, 2011) and stochastic gradient descent (SGD) algorithms. First, let $z(\boldsymbol{\lambda}) := -E[\ell(\boldsymbol{\lambda}; \mathbf{g})]$, and $\boldsymbol{\lambda}^* := \text{argmin}_{\boldsymbol{\lambda}} z(\boldsymbol{\lambda})$. The quantity $\boldsymbol{\lambda}_T^*$ can be viewed as the minimizer of $\frac{1}{T}\sum_{t=1}^T \ell(\boldsymbol{\lambda}; \mathbf{g}^t)$ (the initial loss function can be argued to be negligible after dividing by T). Then Theorem 5.9 in (Pasupathy & Kim, 2011) stipulates that $\sqrt{T}(\boldsymbol{\lambda}_T^* - \boldsymbol{\lambda}^*) \xrightarrow{d} N(0, \Sigma)$, where

$$\Sigma = (\nabla^2 z(\boldsymbol{\lambda}))^{-1} Var(\nabla \ell(\boldsymbol{\lambda}; \mathbf{g}))(\nabla^2 z(\boldsymbol{\lambda}))^{-1} \quad (15)$$

and $Var(\cdot)$ denotes the covariance matrix.

Now since $\nabla^2 L_T(\boldsymbol{\lambda}_T^*; \mathbf{g}^{1:T}) = \sum_{t=1}^T (\nabla^2 \ell(\boldsymbol{\lambda}_T^*; \mathbf{g}^t))$ and $\frac{1}{T}\sum_{t=1}^T (\nabla^2 \ell(\boldsymbol{\lambda}_T^*; \mathbf{g}^t)) \to E[\nabla^2 \ell(\boldsymbol{\lambda}^*; \mathbf{g})]$ by law of large numbers (Durrett, 2010), we have $\nabla^2 L_T(\boldsymbol{\lambda}_T^*; \mathbf{g}^{1:T}) = \Theta(T)$. Then the bound in (5) implies that $|E_{\boldsymbol{\lambda}_T|\mathbf{g}^{1:T}}[\boldsymbol{\lambda}_T] - \boldsymbol{\lambda}_T^*| = o\left(\frac{1}{\sqrt{T}}\right)$. In other words, the difference between posterior mean and $\boldsymbol{\lambda}_T^*$ is of smaller scale than $1/\sqrt{T}$. By Slutsky Theorem (Serfling, 2009), this implies that $\sqrt{T}(E_{\boldsymbol{\lambda}_T|\mathbf{g}^{1:T}}[\boldsymbol{\lambda}_T] - \boldsymbol{\lambda}^*) \xrightarrow{d} N(0, \Sigma)$ also.

On the other hand, for SGD (6), it is known (e.g. (Asmussen & Glynn, 2007)) that the optimal step size parameter value is $\epsilon_T = 1/T$ and $K = \nabla^2 z(\boldsymbol{\lambda})$, in which case the central limit theorem for the update $\boldsymbol{\lambda}_T$ will be given by $\sqrt{T}(\boldsymbol{\lambda}_T - \boldsymbol{\lambda}^*) \xrightarrow{d} N(0, \Sigma)$ where Σ is exactly (15). For other choices of step size, either the convergence rate is slower than order $1/\sqrt{T}$ or the asymptotic variance, denoted by $\tilde{\Sigma}$, is such that $\tilde{\Sigma} - \Sigma$ is positive definite. Therefore, by comparing the asymptotic variance, the posterior mean always has a faster convergence unless the step size in SGD is chosen optimally. □

A.3. Proof of Theorem 3

Proof. Suppose $\boldsymbol{\lambda}$ is used in the strong classifier (11). Denote $I(\cdot)$ as the indicator function. Consider

$$\begin{aligned}
& E_{(\mathbf{x},y)}\left[\sum_{i=1}^m \lambda_i g_i(\mathbf{x},y)\right] = \int \Big[\sum_{i=1}^m \lambda_i g_i(\mathbf{x},1) \\
& P(y=1|\mathbf{x}) + \sum_{i=1}^m \lambda_i g_i(\mathbf{x},-1) P(y=-1|\mathbf{x})\Big] dP(\mathbf{x}) \\
\geq & \int \Big[I\left(\sum_{i=1}^m \lambda_i g_i(\mathbf{x},1) > \sum_{i=1}^m \lambda_i g_i(\mathbf{x},-1)\right) \cdot \sum_{i=1}^m \lambda_i \\
& g_i(\mathbf{x},1) P(y=1|\mathbf{x}) + I\Big(\sum_{i=1}^m \lambda_i g_i(\mathbf{x},1) < \sum_{i=1}^m \lambda_i \\
& g_i(\mathbf{x},-1)\Big) \cdot \sum_{i=1}^m \lambda_i g_i(\mathbf{x},-1) P(y=-1|\mathbf{x})\Big] dP(\mathbf{x}) \\
\geq & \int \Big[I\left(\sum_{i=1}^m \lambda_i g_i(\mathbf{x},1) > \sum_{i=1}^m \lambda_i g_i(\mathbf{x},-1)\right) \cdot \sum_{i=1}^m \lambda_i \\
& g_i(\mathbf{x},-1) P(y=1|\mathbf{x}) + I\Big(\sum_{i=1}^m \lambda_i g_i(\mathbf{x},1) < \sum_{i=1}^m \lambda_i \\
& g_i(\mathbf{x},-1)\Big) \cdot \sum_{i=1}^m \lambda_i g_i(\mathbf{x},1) P(y=-1|\mathbf{x})\Big] dP(\mathbf{x}) \\
= & E_{(\mathbf{x},y)}\left[I(\text{error}) \sum_{i=1}^m \lambda_i g_i(\mathbf{x},-y)\right] \\
\geq & P(\text{error})^p \left(E_{(\mathbf{x},y)}\left[\left(\sum_{i=1}^m \lambda_i g_i(\mathbf{x},-y)\right)^{\frac{-1}{p-1}}\right]\right)^{-(p-1)}
\end{aligned}$$

the last inequality holds by reverse Holder inequality (Hardy et al., 1952). So

$$\begin{aligned}
P(\text{error}) \leq & \left(E_{(\mathbf{x},y)}\left[\sum_{i=1}^m \lambda_i g_i(\mathbf{x},y)\right]\right)^{\frac{1}{p}} \\
& \cdot \left(E_{(\mathbf{x},y)}\left[\left(\sum_{i=1}^m \lambda_i g_i(\mathbf{x},-y)\right)^{\frac{-1}{p-1}}\right]\right)^{\frac{p-1}{p}}
\end{aligned}$$

and the result (12) follows by plugging in $\lambda_i = \frac{1}{\theta E_{(\mathbf{x},y)}[g_i(\mathbf{x},y)]}$ for each i, the minimizer of $E[\ell(\boldsymbol{\lambda};\mathbf{g})]$, which can be solved directly when ℓ is in the form (7). □

A.4. Proof of Theorem 4

Proof. Since for each i, g_i^t are i.i.d., the sample mean $(1/T)\sum_{t=1}^T \mathbf{g}_i^t$ follows a central limit theorem. It can be argued using the delta method (Serfling, 2009) that the posterior mean (10) satisfies

$$\sqrt{T}\left(\frac{\alpha+T}{\beta+\theta\sum_{t=1}^T g_i^t} - \frac{1}{\theta E[g_i(\mathbf{x},y)]}\right)$$

$$\longrightarrow \quad N\left(0, \frac{Var(g_i(\mathbf{x},y))}{\theta^2 (E[g_i(\mathbf{x},y)])^4}\right) \tag{16}$$

For the stochastic gradient descent scheme (13), it would be useful to cast the objective function as $z_i(\lambda_i) = E[\theta\lambda_i g_i - \log\lambda_i]$. Let $\lambda_i^* = \text{argmin}_\lambda z_i(\lambda)$ which can be directly solved as $\frac{1}{\theta E[g_i]}$. Then $z_i''(\lambda_i^*) = \frac{1}{\lambda_i^{*2}} = \theta^2(E[g_i(\mathbf{x},y)])^2$. If the step size $\gamma > \frac{1}{2z''(\lambda_i^*)}$, the update scheme (13) will generate λ_i^T that satisfies the following central limit theorem (Asmussen & Glynn, 2007; Kushner & Yin, 2003)

$$\sqrt{T}(\lambda_i^T - \lambda_i^*) \xrightarrow{d} N(0,\sigma_i^2) \tag{17}$$

where

$$\sigma_i^2 = \int_0^\infty e^{(1-2\gamma z_i''(\lambda_i^*))s}\gamma^2 Var\left(\theta g_i(\mathbf{x},y) - \frac{1}{\lambda_i^*}\right) ds \tag{18}$$

and $\theta g_i(\mathbf{x},y) - \frac{1}{\lambda_i^*}$ is the unbiased estimate of the gradient at the point λ_i^*. On the other hand, $\lambda_i^T - \lambda_i^* = \omega_p(\frac{1}{\sqrt{T}})$ if $\gamma \leq \frac{1}{2z''(\lambda_i^*)}$, i.e. the convergence is slower than (17) asymptotically and so we can disregard this case (Asmussen & Glynn, 2007). Now substitute $\lambda_i^* = \frac{1}{\theta E[g_i]}$ into (18) to obtain

$$\begin{aligned}
\sigma_i^2 &= \theta^2\gamma^2 Var(g_i(\mathbf{x},y)) \int_0^\infty e^{(1-2\gamma/\lambda_i^*)s} ds \\
&= \frac{\theta^2\gamma^2 Var(g_i(\mathbf{x},y))}{2\gamma/\lambda_i^* - 1} = \frac{\theta^2\gamma^2 Var(g_i(\mathbf{x},y))}{2\gamma\theta^2(E[g_i(\mathbf{x},y)])^2 - 1}
\end{aligned}$$

and let $\gamma = \tilde{\gamma}/\theta^2$, we get

$$\sigma_i^2 = \frac{\tilde{\gamma}^2 Var(g_i(\mathbf{x},y))}{\theta^2(2\tilde{\gamma}(E[g_i(\mathbf{x},y)])^2 - 1)} \tag{19}$$

if $\tilde{\gamma} > \frac{\theta^2}{2z''(\lambda_i^*)} = \frac{1}{2(E[g_i(\mathbf{x},y)])^2}$.

We are now ready to compare the asymptotic variance in (16) and (19), and show that for all $\tilde{\gamma}$, the one in (16) is smaller. Note that this is equivalent to showing that

$$\frac{Var(g_i(\mathbf{x},y))}{\theta^2(E[g_i(\mathbf{x},y)])^4} \leq \frac{\tilde{\gamma}^2 Var(g_i(\mathbf{x},y))}{\theta^2(2\tilde{\gamma}(E[g_i(\mathbf{x},y)])^2 - 1)}$$

Eliminating the common factors, we have

$$\frac{1}{(E[g_i(\mathbf{x},y)])^2} \leq \frac{\tilde{\gamma}^2}{2\tilde{\gamma} - 1/(E[g_i(\mathbf{x},y)])^2}$$

and by re-arranging the terms, we have

$$(E[g_i(\mathbf{x},y)])^2\left(\tilde{\gamma} - \frac{1}{(E[g_i(\mathbf{x},y)])^2}\right)^2 \geq 0$$

which is always true. Equality holds iff $\tilde{\gamma} = \frac{1}{(E[g_i(\mathbf{x},y)])^2}$, which corresponds to $\gamma = \frac{1}{\theta^2(E[g_i(\mathbf{x},y)])^2}$.Therefore, the asymptotic variance in (16) is always smaller than (19), unless the step size γ is chosen optimally. □

References

Asmussen, S. and Glynn, P. W. *Stochastic simulation: Algorithms and analysis*. Springer, 2007.

Babenko, B., Yang, M. H., and Belongie, S. Visual tracking with online multiple instance learning. In *CVPR*, pp. 983–990, 2009a.

Babenko, B., Yang, M. H., and Belongie, S. A family of online boosting algorithms. In *ICCV Workshops*, pp. 1346–1353, 2009b.

Cesa-Bianchi, N. and Lugosi, G. Potential-based algorithms in on-line prediction and game theory. *Machine Learning*, pp. 239–261, 2003.

Chen, C. F. On asymptotic normality of limiting density functions with bayesian implications. *Journal of the Royal Statistical Society*, pp. 540–546, 1985.

Chen, S. T. personal communication, 2013.

Chen, S. T., Lin, H. T., and Lu, C. J. An online boosting algorithm with theoretical justifications. In *ICML*, pp. 1007–1014, 2012.

Cox, D. R. and Hinkley, D. V. *Theoretical Statistics*. Chapman & Hall/CRC, 1974.

Doucet, A. and Johansen, A. M. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, pp. 656–704, 2009.

Durrett, R. *Probability Theory and Examples*. Cambridge Series in Statistical and Probabilistic Mathematics, 4th edition, 2010.

Grabner, H. and Bischof, H. On-line boosting and vision. In *CVPR*, pp. 260–267, 2006.

Grabner, H., Leistner, C., and Bischof, H. Semi-supervised on-line boosting for robust tracking. In *ECCV*, pp. 234–247. 2008.

Grbovic, M. and Vucetic, S. Tracking concept change with incremental boosting by minimization of the evolving exponential loss. In *Machine Learning and Knowledge Discovery in Databases*, pp. 516–532. 2011.

Hardy, G. H., Littlewood, J. E., and Pólya, G. *Inequalities*. Cambridge university press, 1952.

Hoeting, J. A., Madigan, D., Raftery, A. E., and Volinsky, C. T. Bayesian model averaging: a tutorial. *Statistical science*, pp. 382–401, 1999.

Kolter, J. Z. and Maloof, M. A. Dynamic weighted majority: An ensemble method for drifting concepts. *The Journal of Machine Learning Research*, pp. 2755–2790, 2007.

Kolter, J.Z. and Maloof, M.A. Using additive expert ensembles to cope with concept drift. In *Proceedings of the 22nd international conference on Machine learning*, pp. 449–456, 2005.

Kushner, H. J. and Yin, G. *Stochastic approximation and recursive algorithms and applications*. Springer, 2003.

Laarhoven, V., JM, P., and Aarts, E. H. *Simulated annealing*. Springer, 1987.

Leistner, C., Saffari, A., Roth, P. M, and Bischof, H. On robustness of on-line boosting-a competitive study. In *ICCV Workshops*, pp. 1362–1369, 2009.

Liu, X. and Yu, T. Gradient feature selection for online boosting. In *ICCV*, pp. 1–8, 2007.

Minku, L.L. *Online ensemble learning in the presence of concept drift*. PhD thesis, University of Birmingham, 2011.

Oza, N. C. *Online ensemble learning*. PhD thesis, University of California, Berkeley, 2001.

Oza, N. C. and Russell, S. Online bagging and boosting. In *AISTATS*, pp. 105–112, 2001.

Pasupathy, R. and Kim, S. The stochastic root-finding problem: overview, solutions, and open questions. *ACM Transactions on Modeling and Computer Simulation*, 21 (3):19, 2011.

Pelossof, R., Jones, M., Vovsha, I., and Rudin, C. Online coordinate boosting. In *ICCV Workshops*, pp. 1354–1361, 2009.

Saffari, A., Godec, M., Pock, T., Leistner, C., and Bischof, H. Online multi-class lpboost. In *CVPR*, pp. 3570–3577, 2010.

Schapire, Robert E. Drifting games. *Machine Learning*, pp. 265–291, 2001.

Serfling, R. J. *Approximation theorems of mathematical statistics*. Wiley. com, 2009.

Wang, H., Fan, W., Yu, P.S., and Han, J. Mining concept-drifting data streams using ensemble classifiers. In *SIGKDD*, pp. 226–235, 2003.