Cost-sensitive Boosting with p-norm Loss Functions and its Applications

Aurélie C. Lozano
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598
aclozano@us.ibm.com

Naoki Abe
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598
nabe@us.ibm.com

Abstract

In practical applications of classification, there are often varying costs associated with different types of misclassification (e.g. fraud detection, anomaly detection and medical diagnosis), motivating the need for the so-called ”cost-sensitive” classification. In this paper, we introduce a family of novel boosting methods for cost-sensitive classification by applying the theory of gradient boosting to p-norm based cost functionals, and establish theoretical guarantees as well as their empirical advantage over existing algorithms.

1 Introduction

Many real world applications of classification techniques involve dramatically varying costs associated with different types of misclassification, motivating the need for classification algorithms that pay attention to the “cost” of misclassification. Relevant application areas include credit rating, targeted marketing, fraud detection, anomaly detection in manufacturing processes and medical diagnosis, among others. For example, in credit rating, the cost of misclassifying a good customer as bad is the opportunity cost of missing the associated profit, whereas the cost of misclassifying a bad customer as good is the cost of default, which is generally much higher. Furthermore, these two types of costs actually depend on the individual customer, namely, depending on their amount of loan. There has been considerable theoretical as well as empirical research on this topic, which is known as “cost sensitive learning”, both in the machine learning and data mining communities [5, 3, 13, 15, 16].

For pure classification, extensive past research has established that the family of boosting methods, including AdaBoost [6] and its many variations, enjoys superior empirical performance and strong theoretical guarantees. For cost-sensitive learning, however, there has not been a comprehensive study of relative merits of different boosting algorithms. Some attempts have been made to extend the AdaBoost algorithm into cost-sensitive versions, e.g. AdaCost [5] and CSB2 [13], but the aggressive weight updating scheme based on the exponential loss posed difficulties in balancing the contributions of the cost information and boosting’s focus on misclassification error. More recently, an effort was made to bridge this gap with the proposal of a cost-sensitive boosting method called GBSE [1], inspired by the framework of gradient boosting, but only a partial theoretical justification was provided, where the proof of convergence was given for a variant of the proposed method.

In this paper, we propose a class of new cost-sensitive boosting methods by applying the theory of gradient boosting to a family of p-norm cost functionals, and investigate their theoretical and empirical properties. The p-norm cost functionals include, as special cases, the linear cost (expected cost) and the squared loss based cost functional. We derive a family of general multi-class cost-sensitive boosting methods, which use binary weak classifiers, and establish some basic properties for this family, including proof of convergence and the rate thereof. We also give interpretations for some of the existing algorithms in terms of the proposed family, notably including a generalization of the costing algorithm [16], DSE and GBSE [1], and the Average Cost method [8].

We empirically compare the performance of the proposed family of algorithms with representative existing methods of cost-sensitive boosting, including AdaCost [5], CSB2 [13] as well as AdaBoost.M2 [6] with cost-sensitive weight initialization. Our experiments, using multi-class data sets from the UCI repository, indicate that the proposed methods based on p-norm cost functionals attain excellent performance in terms of cost minimization. More specifically, the generalized costing algorithm for the linear loss function compares favorably against AdaCost and CSB2, but does not out-perform AdaBoost with cost-sensitive weight initialization. In contrast, the variants with higher order p-norm loss functions, including squared loss, strictly out-perform all three of the comparison methods, establishing the advantage of the proposed family of methods.

2 Preliminaries

We first describe the framework for cost sensitive learning along with related concepts and notation used in this paper.

We consider a general formulation of cost-sensitive (multiclass) classification[1] where a cost function $C(x, y_1, y_2)$ is used to specify the cost of predicting that an example x belongs to class y_2 when the correct label is y_1. Formally, denote by X the input space and by Y the set of classes. Let $k = |Y|$. We assume that examples $(x, \vec{C})$ are drawn from a distribution D over $X \times {R^+}^k$. Here $\vec{C}$ is the vector of costs $C_{x,y} = C(x, y, y^*)$ where y^* denotes the label with minimum cost and $y \in Y$. We note that the above formulation allows C to depend on the individual instances, x, generalizing the common formulation in terms of cost matrices, following [15].

Based on a sample $S = \{(x, \vec{C})\}$ drawn i.i.d. from D, we wish to find a classifier $h : X \to \{1, \dots, k\}$ which minimizes the expected cost

$$\mathrm{E}_{(x,\vec{C})\sim D}[C_{x,h(x)}].$$

Without loss of generality we assume that the costs are normalized so that $\forall x \in X \; C_{x,y^*} = C(x, y^*, y^*) = 0$. Then the problem is equivalent to the minimization in term of misclassification cost, i.e.

$$\arg\min_{h\in\mathcal{H}} \mathrm{E}_{(x,\vec{C})\sim D}[C_{x,h(x)} I(h(x) \neq y^*)],$$

where $I(\cdot)$ denotes the indicator function.

Our proposed methods will make use of *importance weighted classification*, which we review below. In importance-weighted classification, examples of the form (x, y, w) are drawn from a distribution D over $X \times Y \times R^+$. Given a training set $S = \{(x, y, w)\}$ the goal is to find a classifier $h : X \to Y$ having minimum expected weighted misclassification error:

$$E_{(x,y,w)\sim D}\left[w \cdot I(h(x) \neq y)\right].$$

3 Methodology

3.1 Minimization of a Convex Objective: a p-norm Based Cost Functional

The aforementioned concepts were introduced in terms of *functional* hypotheses h, i.e. $h : X \to Y$. but also apply to *stochastic* hypotheses, namely hypotheses $h : X \times Y \to [0, 1]$ satisfying the stochastic condition $\forall x \in X \; \sum_{y\in Y} h(y|x) = 1$. In particular a stochastic cost-sensitive learner is expected to minimize

$$E_{(x,\vec{C})\sim D}[C_{x,\arg\max_y h(y|x)}].$$

Since D is unknown, one could consider methods which, given a training sample, attempt to minimize the sample average

$$\hat{E}_{(x,\vec{C})\sim S}[C_{x,\arg\max_y h(y|x)}] = \frac{1}{|S|} \sum_{(x,\vec{C})\in S} C_{x,\arg\max_y h(y|x)}. \quad (1)$$

[1] This general formulation allowing costs to depend on individual instances was first proposed by Zadrozny & Elkan [15].

Such procedures, however, are computationally intractable for many hypothesis classes, one major obstacle being the non-convexity of the objective. To remedy this issue, the learning methods proposed in this paper are based on the minimization of a convex surrogate of the objective. Notice that $\max_y h(y|x) = \|h(y|x)\|_\infty$. Thus

$$C_{x,\arg\max_y h(y|x)} = \sum_y \left(\frac{h(y|x)}{\|h(y|x)\|_\infty} \right)^\infty C_{x,y},$$

which can be approximated by

$$\sum_y \left(\frac{h(y|x)}{\max_{y'} h(y'|x)} \right)^p C_{x,y}, \ \ p \geq 1.$$

Since $\max_y h(y|x) \geq 1/k$, it is natural to consider the minimization of the following convexification of the original objective of Eq. 1:

$$\arg\min_h \frac{1}{|S|} \sum_{(x,\vec{C}) \in S} \sum_y (h(y|x))^p \, C_{x,y}, \ \ p \geq 1 \tag{2}$$

We expect that the larger p is the closer it approximates Eq. 1.

3.2 Cost Sensitive Boosting with p-norm Loss

The minimization of the convex objective as defined in Eq. 2 is carried out by adopting a boosting-style functional gradient descent approach (see [10, 9]) combined with a stochastic interpretation of ensemble hypotheses. We now elaborate on our methodology.

Given multiple functional hypotheses $h_t, \ t = 1, \ldots, T$, we define a stochastic ensemble hypothesis H as the conditional distribution resulting from the mixture of the component hypotheses, namely,

$$\forall x \in X, \forall y \in Y \ H(y|x) = \frac{1}{T} \sum_{t=1}^{T} I(h_t(x) = y).$$

To solve Eq. 2, an incremental algorithm is used, which, at each round t, updates the current ensemble hypothesis by the convex combination of the previous ensemble hypothesis H_{t-1} and a new hypothesis h_t, i.e., by setting

$$H_t(y|x) = (1 - \beta) H_{t-1}(y|x) + \beta I(h_t(x) = y),$$

where $\beta \in [0, 1]$.

Let $f_t(y|x) = I(h_t(x) = y)$ and let ∇ denote the gradient operator. The new hypothesis h_t is output by a weak learner so as to maximize $-\langle \nabla L(H_{t-1}), f - H_{t-1} \rangle$, where

$$L(H) = \sum_{x \in X} \sum_{y \in Y} H(y|x)^p C_{x,y}.$$

We remark that Mason et al. [10] also used this type of gradient descent formulation. By the Fréchet-like differentiability condition of the p-norm cost functional considered

$$\begin{aligned}
&\langle \nabla L(H), f - H \rangle \\
&= \lim_{\alpha \to 0^+} \frac{L\left((1-\alpha)H + \alpha f\right) - L(F)}{\alpha} \\
&= \sum_{x,y} p H(y|x)^{p-1} C_{x,y} (f(y|x) - H(y|x)).
\end{aligned} \tag{3}$$

So at each iteration t, h_t is chosen to minimize

$$\sum_{x \in X} \sum_{y \in Y} w_{x,y} (I(h_t(x) = y)), \tag{4}$$

where

$$w_{x,y} = H_{t-1}(y|x)^{p-1} C_{x,y}. \tag{5}$$

This optimization problem with respect to these particular weights is the basis of the family of methods proposed in this paper. Note, in particular, that the special case in which $p = 1$, the weights become $w_{x,y} = C_{x,y}$, corresponding to various existing methods with cost based weighting, including the costing algorithm [16] and DSE [1].

3.3 Learning Methods

We now consider implementing the optimization problem of Eq. 4 using *binary* base learning procedures. While it is also possible to use a multi-class classifier as the base learner, in practice it tends not to work as well as those based on binary classifiers, hence our focus on binary weak learners. When converting multiclass into binary classification, it is useful to consider the notion of *relational* hypotheses, namely those that are relations over $X \times Y$: $h : X \times Y \rightarrow \{0, 1\}$.

A straightforward multiclass method to find a new weak hypothesis consists in using a weak learner minimizing the weighted classification error for the expanded data set

$$S' = \Big\{(x, y)|\exists(x, \vec{C}) \in S \text{ and } y \in Y\Big\}$$

and weights

$$\max_{y'} w_{x,y'} - w_{x,y}.$$

(It was shown in [1] that this also minimizes Eq. 4.) We remark that for linear loss ($p = 1$), the resulting procedure is identical to the DSE Method [1], as the weights reduce to $\max_{y'} C_{x,y'} - C_{x,y}$, which are constant over the iterations. Notice that with this expanded dataset formulation all the labels are effectively treated as correct labels, albeit to varying degrees due to having different weights, resulting in sub-optimal performance in practice. In subsequent developments, we address this issue in a number of different ways, resulting in various concrete algorithms implementing the weighting scheme of Eq. 4.

3.3.1 L_p-Cost Sensitive Boosting

Consider again the optimization problem of Eq. 4, and the corresponding weights in Eq. 5. Notice that $C_{x,y^*} = 0$ implies $w_{x,y^*} = 0$, and hence the optimization problem of Eq. 4 effectively involves the dataset

$$B = \{(x, y)|x \in X, y \in Y, y \neq y^*\}\,. \tag{6}$$

For *stochastic* hypotheses, (x, y^*) is indirectly taken into account, since for any such hypothesis f, $\forall x\, f(y^*|x) = 1 - \sum_{y \in Y, y \neq y^*} f(y|x)$. For *relational* hypotheses $h : X \times Y \rightarrow \{0, 1\}$, however, the minimization of Eq. 4 can be achieved simply by assigning $h(x, y) = 0$ everywhere. The pseudo-loss [6] is thus introduced as a way to explicitly incorporate $h(x, y^*)$ in the objective:

$$\frac{1}{2} \sum_{(x,y) \in B} w_{x,y}(1 - h(x, y^*) + h(x, y)).$$

We can reformulate this minimization problem as a weighted binary classification problem by converting the weighted sample $\{(x, y, w_{x,y}), x \in X, y \in Y\}$ into

$$S_2 = \Big\{((x, y), l, w'_{x,y}), x \in X, y \in Y\Big\}\,,$$

where

$$\begin{cases} w'_{x,y} = \frac{w_{x,y}}{2} \text{ and } l = 0 & \forall (x, y) \in B \\ w'_{x,y^*} = \frac{\sum_{y \neq y^*} w_{x,y}}{2} \text{ and } l = 1 & \forall x \in X \end{cases}$$

The component learner is then to find a relational hypothesis h that minimizes the weighted error on S_2, i.e.,

$$\sum_{x \in X} \sum_{y \in Y} w'_{x,y} I(h(x, y) \neq l),$$

which is equivalent to minimizing the pseudo loss. The resulting procedure, L_p-CSB (Cost Sensitive Boosting with p-norm Loss) is our main method and is depicted in Figure 1.

```
• An input sample $S = \{(x, \vec{C})\}$.
• A component learner $A$ for importance weighted binary classification that takes a sample of the form $((x, y), l, w)$ and output a relational hypothesis $h$.
• An integer $T$ specifying the number of iterations to be performed.
1. Set $S' = \left\{(x, y)|(x, \vec{C}) \in S, y \in Y\right\}$.
2. Initialize $H_0$ by $\forall x \in X, y \in Y H_0(y|x) = 1/k$.
3. **For** t:=1 to T **Do**
    (a) $w_{x,y} = H_{t-1}(y|x)^{p-1} C_{x,y}$ for all $(x, y)$ in $S'$.
    (b) For all $(x, y)$ in $S'$ such that $y \neq y^*$ $w'_{x,y} = w_{x,y}/2$ and $l_{x,y} = 0$.
        For all $(x, y^*)$ in $S'$ $w'_{x,y^*} = (\sum_{y \neq y^*} w_{x,y})/2$ and $l_{x,y^*} = 1$.
    (c) $S_t = \left\{((x, y), l_{x,y}, w'_{x,y})|(x, y) \in S'\right\}$.
    (d) Let $h_t := A(S_t)$.
    (e) Choose $\alpha_t \in [0, 1)$, for example $\alpha_t = \frac{1}{t}$.
    (f) Set $H_t := (1 - \alpha_t)H_{t-1} + \alpha_t h_t$.
4. **End For**
5. Return $H_T$.
```

Figure 1: Method L_p-*CSB (Cost Sensitive Boosting with* p*-norm Loss)*

3.3.2 Relationship Between Pseudo-Loss and Original Loss

We now discuss several ways to characterize the relationship between the pseudo-loss and the original loss

$$l_w(h) = \sum_{x \in X, y \in Y} w_{x,y} h(x, y).$$

For convenience we omit the factor $1/2$ in the pseudo-loss and let

$$\tilde{l}_w(h) = \sum_{(x,y) \in B} w_{x,y}(1 - h(x, y^*) + h(x, y)),$$

since the minimizer is unaffected. Notice that

$$\tilde{l}_w(h) = l(h) + \sum_{x \in X}((\sum_{y \in Y} w_{x,y})(1 - h(x, y^*))). \tag{7}$$

Hence if for some hypothesis h, $\tilde{l}_w(h) = \epsilon$ then $l_w(h) < \epsilon$. So an hypothesis with small pseudo-loss has small original loss as well.

For stochastic hypotheses, the equivalence between pseudo loss and original loss is expressed by the following proposition.

Proposition 3.1. *For any stochastic hypothesis* $h : X \times Y \to \{0, 1\}$, $l_w(h) = \tilde{l}_{\tilde{w}}(h)$ *if*

$$\forall (x, y) \in B \;\; \tilde{w}_{x,y} = w_{x,y} - \frac{1}{k} \sum_{y' \neq y^*} w_{x,y'}$$

where B *is as defined in Eq. 6.*

Proof.

$$\begin{aligned}
\tilde{l}_{\tilde{w}}(h) &= \sum_x \sum_{y \neq y^*} \tilde{w}_{x,y}(1 + h(x, y) - h(x, y^*)) \\
&= \sum_x \sum_{y \neq y^*} \tilde{w}_{x,y} h(x, y) + \sum_x \left(\sum_{y \neq y^*} \tilde{w}_{x,y} \right) (\sum_{y \neq y^*} h(x, y)) \qquad (8) \\
&= \sum_x \sum_{y \neq y^*} \left(\tilde{w}_{x,y} + \sum_{y' \neq y^*} \tilde{w}_{x,y'} \right) h(x, y). \qquad (9)
\end{aligned}$$

```
• An input sample $S = \{(x, \vec{C})\}$.
• A component learner $A(\tilde{S})$ that takes a training sample $\tilde{S} = \{(x, y, w_{x,y})\}$ , and output a
stochastic hypothesis $f$ that attempts to minimize $\sum_{(x,y)} w_{x,y} f(y|x)$.
• An integer $T$ specifying the number of iterations to be performed.
1. Set $S' = \left\{(x, y)|(x, \vec{C}) \in S, y \in Y\right\}$ .
2. Initialize $H_0$ by $\forall x \in X, y \in Y H_0(y|x) = 1/k$.
3. For t:=1 to T Do
    (a) $w_{x,y} = H_{t-1}(y|x)^{p-1} C_{x,y}$ for all $(x, y)$ in $S'$.
    (b) $S_t = \{(x, y, w_{x,y})|(x, y) \in S'\}$ .
    (c) Let $f_t := A(S_t)$.
    (d) Choose $\alpha_t \in [0, 1)$
    (d) Set $H_t := (1 - \alpha_t)H_{t-1} + \alpha_t f_t$.
4. End For
5. Return $H_T$.
```

Figure 2: Method L_p-*CSB-A (Abstracted view of Cost Sensitive Boosting with p-norm Loss).*

Here Eq. 8 follows from the fact that h is stochastic and hence $1 - h(x, y^*) = \sum_{y \neq y^*} h(x, y)$. Now if one sets $w_{x,y} = \tilde{w}_{x,y} + \sum_{y' \neq y^*} \tilde{w}_{x,y'}$ for $(x, y) \in B$, the right-hand side of Eq. 9 equals $l_w(h)$, since for all $x \in X \;\; w_{x,y^*} = 0$. Now setting $w_{x,y} = \tilde{w}_{x,y} + \sum_{y' \neq y^*} \tilde{w}_{x,y'}$ for all $(x, y) \in B$ implies $\tilde{w}_{x,y} = w_{x,y} - \frac{1}{k} \sum_{y' \neq y^*} w_{x,y'}$ for $(x, y) \in B$, since $\sum_{y' \neq y^*} \tilde{w}_{x,y'} = \frac{1}{k} \sum_{y' \neq y^*} w_{x,y'}$. □

3.3.3 L_p-Cost Sensitive Boosting with Pseudo-Loss Adjustment

Proposition 3.1 naturally suggests an alternative method, L_p-CSB-PA (Cost Sensitive Boosting with p-norm Loss and Pseudo-loss Adjustment), which is similar to L_p-CSB but where the weights $w_{x,y}$ are replaced by $\tilde{w}_{x,y}$, and stochastic hypotheses are required.

Note that, for the linear loss ($p = 1$), the weights $\tilde{w}_{x,y}$ are identical to those of the "Average Cost" method due to Margineantu [8]. Also, when $p = 2$, that is under squared loss, the weights become

$$\tilde{w}_{x,y} = H_{t-1}(y|x)C_{x,y} - \frac{1}{k} C_{H_{t-1}}(x),$$

where

$$C_{H_{t-1}}(x) = \sum_{y \in Y} H_{t-1}(x, y)C_{x,y}.$$

These weights are remarkably similar (*though different*) to the weighting scheme of the GBSE-t algorithm [1] in the sense that for each sample (x, y) both weighting schemes involve relating the cost at (x, y) to the average cost incurred by the current hypothesis at x divided by the number of classes.

Notice that the weights $\tilde{w}_{x,y}$ can be negative, which implies that the component learner is asked to minimize weighted misclassification with positive and negative weights. This is a perfectly valid optimization problem, but implementing it using a standard weak learner requires a transformation: converting each example $((x, y), 1, w'_{x,y})$ for which $w'_{x,y} < 0$ into $((x, y), 0, -w'_{x,y})$.

4 Convergence Analysis

For simplicity we consider the "abstracted" version of our methods depicted in Figure 2, which is expressed in terms of the original optimization problem stated in Eq. 4. Denote by $\mathcal{F}$ the class of base hypotheses and by $\mathcal{H}$ the set of convex combinations of hypotheses in $\mathcal{F}$.

At each round t, the new hypothesis f_t returned by the weak learner attempts to minimize $\sum_{x,y} w_{x,y} f(y|x)$, with $w_{x,y} = H_{t-1}(y|x)^{p-1} C_{x,y}$. The following theorem characterizes convergence in terms of the relative performance of the selected weak hypothesis in completing that task compared to that of the current composite hypothesis H_{t-1}.

Theorem 4.1. *Consider the method L_p-CSB-A. Assume that at each iteration t the new hypothesis f_t returned by the weak learner is such that*

$$\sum_{x,y} w_{x,y}\left(f_t(y|x) – H_{t-1}(y|x)\right) \leq -\epsilon_t,$$

with $\epsilon_t \geq 0$. Pick

$$\alpha_t = -\frac{\sum_{x,y} pH_{t-1}(y|x)^{p-1}C_{x,y}(f_t(y|x) – H_{t-1}(y|x))}{M\|f_t - H_{t-1}\|^2}, \tag{10}$$

where $M = \sup_{x\in X, y\in Y} C_{x,y}p(p-1)(2^{p-2})$. Then the algorithm converges to the global minimum of the cost L over $\mathcal{H}$.

Proof. Notice that L is Lipschitz differentiable with Lipschitz constant M, i.e. $\|\nabla L(H_2) - \nabla L(H_1)\| \leq M\|H_2 - H_1\|$, for all H_1, H_2 in $\mathcal{H}$. Hence

$$\begin{aligned} L((1-\alpha)H + \alpha f) - L(H) &= L(H + \alpha(f - H)) - L(H) \\ &\leq \alpha\langle\nabla L(H), f - H\rangle + \frac{M\alpha^2}{2}\|f - H\|^2, \end{aligned}$$

where the inequality is obtained by applying Lemma 3 of [10]. This upperbound can be optimized by setting:

$$\alpha = -\frac{\langle\nabla L(H), f - H\rangle}{M\|f - H\|^2}.$$

We thus have for this choice of α

$$\begin{aligned} L((1-\alpha)H + \alpha f) - L(H) &\leq -\frac{1}{2}\frac{\langle\nabla L(H), f - H\rangle^2}{M\|f - H\|^2} \\ &\leq -\frac{1}{8}\frac{\langle\nabla L(H), f - H\rangle^2}{M}, \end{aligned} \tag{11}$$

where the last inequality follows from the fact that $\|f - H\|^2 \leq 4$, since H and f are defined on $[0, 1]$.
Combining Eq. 11 and Eq. 3 we obtain

$$L(H_t) - L(H_{t-1}) \leq \frac{-p}{8M}\left(\sum_{x,y} w_{x,y}\left(f_t(y|x) – H_{t-1}(y|x)\right)\right)^2,$$

with $w_{x,y} = H_{t-1}(y|x)^{p-1}C_{x,y}$. Hence

$$L(H_t) - L(H_{t-1}) \leq \frac{-p}{8M}\epsilon_t^2. \tag{12}$$

Assume that if at a certain round t' $\sum_{x,y} w_{x,y}\left(f_{t'}(y|x) – H_{t'-1}(y|x)\right) = 0$ (i.e. $\epsilon_{t'} = 0$) then the algorithm returns $H_{t'-1}$ for all $t \geq t'$. Since L is lower bounded, this and Eq. 12 imply that the sequence H_t converges to an accumulation point H. We conclude by using the proof of Theorem 6 of [10], which shows that any such accumulation point has minimum cost L over $\mathcal{H}$. □

The weak learning assumption required by the theorem is reasonable as f_t is specifically picked as an attempt to minimize $\sum_{x,y} w_{x,y}f(y|x)$, while H_{t-1} is not, and hence it is reasonable to expect the former to outperform the latter. Note that the weak learning assumption is similar to what is asked of the weak learner in MarginBoost.L_1 [10], in the sense that it involves the weighted average of the difference between current composite classifier and weak hypothesis.

The next theorem considers approximate minimization of $\sum_{(x,y)} w_{x,y}\left(f_t(y|x)\right)$ over the prescribed class $\mathcal{F}$ of weak hypotheses as the weak learning condition, and provide some convergence rates for the L_p-CSB-A procedure.

Theorem 4.2. *Assume that at each iteration t of the algorithm L_p-CSB-A the component learner returns hypothesis f_t such that*

$$\sum_{(x,y)} w_{x,y} f_t(y|x) \leq \inf_f \sum_{(x,y)} w_{x,y} f(y|x) + \epsilon_t, \tag{13}$$

where $\epsilon_t \leq \frac{M}{2(t+1)^2}$. Pick α_t as in Eq. 10. Then for all $t > 0$ we have for H_t obtained by L_p-CSB-A that

$$L(H_t) - \inf_{H \in \mathcal{H}} L(H) \leq \frac{9M}{(t+2)}, \tag{14}$$

where $M = \sup_{x \in X, y \in Y} C_{x,y} p(p-1)(2^{p-2})$.

Proof. From the proof of Theorem 4.1, we have

$$L((1-\alpha)H + \alpha f) - L(H) \leq -\frac{1}{8} \frac{\langle \nabla L(H), f - H \rangle^2}{M}. \tag{15}$$

By convexity of L, $-\langle \nabla L(H), f - H \rangle \geq L(H) - L(f)$.
Thus

$$\sup_{f \in \mathcal{H}} -\langle \nabla L(H), f - H \rangle \geq \sup_{f \in \mathcal{H}} L(H) - L(f). \tag{16}$$

Using Eq. 3 and Eq. 5, notice that the weak learning assumption of Eq. 13 is equivalent to assuming that f_{t+1} is such that

$$\begin{aligned} -\langle \nabla L(H_t), f_{t+1} - H_t \rangle &\geq \sup_{f \in \mathcal{F}} -\langle \nabla L(H_t), f - H_t \rangle - p\epsilon_{t+1} \\ &= \sup_{f \in \mathcal{H}} -\langle \nabla L(H_t), f - H_t \rangle - p\epsilon_{t+1}, \end{aligned} \tag{17}$$

for some $\epsilon_{t+1} \geq 0$. Here we used the fact that the supremum of a linear function over a convex polygon is achieved at one of the vertices. Let $L^+(H) = L(H) - \inf_{f \in \mathcal{H}} L(f)$. Combining Eq. 15, Eq. 16 and Eq. 17 leads to

$$\begin{aligned} L^+(H_{t+1}) &\leq L^+(H_t) - \frac{1}{8M}(L^+(H_t))^2 + \frac{2p\epsilon_{t+1}}{8M}(L^+(H_t)) - \frac{p^2\epsilon_{t+1}^2}{8M} \\ &\leq L^+(H_t) - \frac{1}{8M}(L^+(H_t))^2 + 2\epsilon_{t+1}, \end{aligned} \tag{18}$$

where the last inequality follows from the fact that $L^+(H_t) \leq \frac{4}{p-1}M$, since it can easily be show that $|L(a) - L(b)| \leq p2^{p-1}|a - b|$, for $a, b \in [0, 1]$. An upperbound for $L^+(H_{t+1})$ can be obtained by maximizing the right hand side of Eq. 18 with respect to $L^+(H_t)$. We obtain

$$\forall t \geq 0 \;\; L^+(H_{t+1}) \leq 2M + 2\epsilon_{t+1}. \tag{19}$$

The remainder of the proof consists in showing that Eq. 14 holds for all t. We do so by induction similarly to the proof of Theorem IV.2 in [17]. For $t = 1$ Eq. 19 implies $L^+(H_1) \leq 2M + 2\epsilon_1 \leq 2M + \frac{M}{4} < \frac{9M}{3}$. Assume that $L^+(H_t) \leq \frac{9M}{t+2}$. Note that the function $x \to x - \frac{x^2}{8M}$ is increasing on $[0, 4M]$, and that $L^+(H_t) \leq \frac{9M}{t+2} \leq 4M$. Thus using Eq. 18, we get

$$\begin{aligned} L^+(H_{t+1}) &\leq \frac{9M}{t+2} - \frac{1}{8M}\frac{81M^2}{(t+2)^2} + \frac{M}{(t+2)^2} \\ &= \frac{9M}{t+2} - \frac{73M}{8}\frac{1}{(t+2)^2} \leq \frac{9M}{t+3}. \end{aligned}$$

□

Data Set	AdaBoost	AdaCost	CSB2	Linear Loss	Sq Loss
Splice	34.62 ± 5.0	36.03 ± 3.4	527.62 ± 3.1	34.85 ± 2.1	**31.31 ± 1.9**
Anneal	3086.18 ± 201.4	1964.48 ± 152.9	2231.52 ± 473.6	26.57 ± 2.8	**24.62 ± 3.2**
Sat	124.44 ± 7.1	94.32 ± 8.8	366.26 ± 16.9	84.22 ± 6.3	**77.32 ± 5.5**
Flare	6030.33 ± 360.0	6030.33 ± 360.0	3710.4 ± 943.6	26.71 ± 8.5	**15.99 ± 1.2**
Letter	619.07 ± 28.5	613.89 ± 27.56	614.45 ± 25.11	621.35 ± 27.07	**549.85 ± 64.38**
Pendigits	41.9 ± 2.4	72.04 ± 6.2	68.44 ± 8.8	47.74 ± 2.3	**35.27 ± 1.7**
Segment	26.88 ± 2.0	48.94 ± 5.9	109.55 ± 5.2	30.58 ± 1.9	**16.05 ± 1.6**
Thyroid	$310K \pm 36K$	$311K \pm 37K$	5778 ± 2395	139.75 ± 36.6	**99.48 ± 18.2**

Table 1: Results for comparison methods and the proposed method (with Squared Loss) on the multi-class datasets with the class frequency cost model: the average cost and standard error.

Data Set	Linear Loss	Sq Loss	Cub Loss	Quad Loss	5th Loss
Splice	34.85 ± 2.1	31.31 ± 1.9	30.11 ± 1.8	**30.40 ± 2.0**	31.04 ± 1.6
Anneal	26.57 ± 2.8	24.62 ± 3.2	**23.94 ± 3.2**	29.12 ± 4.1	26.83 ± 3.1
Sat	84.22 ± 6.3	77.32 ± 5.5	**72.68 ± 6.1**	73.25 ± 6.0	73.31 ± 5.7
Flare	26.71 ± 8.5	15.99 ± 1.2	15.99 ± 1.2	15.11 ± 1.1	**15.09 ± 1.1**
Letter	621.35 ± 27.07	**549.85 ± 64.38**	609.47 ± 28.07	610.99 ± 29.15	612.57 ± 28.29
Pendigits	47.74 ± 2.3	35.27 ± 1.7	32.76 ± 1.7	30.85 ± 1.7	**30.66 ± 1.7**
Segment	30.58 ± 1.9	16.05 ± 1.6	16.06 ± 2.9	19.05 ± 2.4	**14.95 ± 2.3**
Thyroid	139.75 ± 36.6	**99.48 ± 18.2**	109.54 ± 9.3	139.61 ± 36.6	125.99 ± 24.7

Table 2: Results for the proposed methods with various p-norm losses on the multi-class datasets with the class frequency cost model: the average cost and standard error.

5 Experiments

We conducted systematic experiments to compare the performance of the proposed methods with a number of existing algorithms: AdaBoost.M2 with cost-sensitive weight initialization, AdaCost and CSB2,[2] using multi-class data sets from the UCI repository.

All of our experiments were run using randomly generated cost matrices, where we basically followed one of the the cost matrix generation procedures considered in [3], with modifications employed in [1]. In our cost model, which we call the *class frequency cost model*, the rare classes tend to be assigned proportionally higher costs than the frequent classes.

As data sets, we elected to use those data sets from UCI ML data repository [2] that (i) are multi-class data sets; and (ii) have a large enough data size (exceeding approximately 1,000). As the "weak" learner, we use the weka j48 implementation of C4.5 in all of our experiments [11, 14]. Other details of the experimental procedures can be found in [7].

Table 1 and Table 2 summarize the results or our experiments, giving the average test set cost and standard error for each of the 8 data sets, and for each of four methods considered.[3] Table 1 compares the average costs of all comparison methods, including a generalization of the costing algorithm (L_p-CSB with $p = 1$), with L_p-CSB with squared loss as the representative of the proposed family of methods. From these results, it appears convincing that the L_p-CSB family of boosting methods out-perform all of the comparison methods we consider. It is interesting to note that the linear case (generalization of costing) does not consistently outperform AdaBoost. We need the squared loss or higher p-norm, which both correspond to boosting weights that vary over iterations, to do so. Table 2 compares the performance of the L_p-CSB family for different values of p. It is seen that for many datasets, the performance continues to improve for higher values of p, which is what would be expected by our motivation to approximate the objective cost function by a convex p-norm functional.

[2] We refer the reader to [7] for the detailed description of the comparison methods we use.

[3] In these and subsequent tables, the figures that correspond to best performance are shown in bold font.

6 Concluding Remarks

We have proposed a novel family of cost-sensitive boosting methods based on p-norm cost functionals and the gradient boosting framework. Our theoretical development provides a framework to interpret a variety of existing methods of cost-sensitive learning and their variants. In addition, we have provided empirical evidence that our approach can lead to excellent performance in practice, as the proposed family of methods outperforms representative algorithms on benchmark data sets.

References

[1] N. Abe, B. Zadrozny, and J. Langford. An iterative method for multi-class cost-sensitive learning. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 3–11, New York, NY, USA, 2004. ACM.

[2] C. L. Blake and C. J. Merz. UCI repository of machine learning databases. Department of Information and Computer Sciences, University of California, Irvine, 1998. `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

[3] P. Domingos. MetaCost: A general method for making classifiers cost sensitive. In *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining*, pages 155–164. ACM Press, 1999.

[4] Charles Elkan. Magical thinking in data mining: Lessons from coil challenge 2000. In *Proceedings of the Seventh International Conference on Knowledge Discovery and Data Mining*, pages 426–431. ACM Press, 2001.

[5] W. Fan, S. J. Stolfo, J. Zhang, and P. K. Chan. AdaCost: Misclassification cost-sensitive boosting. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pages 97–105, 1999.

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

[7] A.C. Lozano and N. Abe Multi-class Cost-sensitive Boosting with p-norm Loss Function. In *KDD '08: Proceedings of the fourteenth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM Press, 2008.

[8] D. Margineantu. *Methods for Cost-Sensitive Learning*. PhD thesis, Department of Computer Science, Oregon State University, Corvallis, 2001.

[9] L. Mason, J. Baxter, P. Barlett, and M. Frean. Boosting algorithms as gradient descent. In *Advances in Neural Information Processing systems 12*, pages 512–158, 2000.

[10] L. Mason, J. Baxter, P. Bartlett, and M. Frean. Boosting algorithms as gradient descent in function space. Technical report, RSISE, Australian National University, 1999. `http://wwwsyseng.anu.edu.au/~jon/papers/doom2.ps.gz`.

[11] J. Quinlan. *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann, 1993.

[12] R. E. Schapire and Y. Singer. Improved boosting using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

[13] K. M. Ting. A comparative study of cost-sensitive boosting algorithms. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 983–990, 2000.

[14] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, 2000.

[15] B. Zadrozny and C. Elkan. Learning and making decisions when costs and probabilities are both unknown. In *Proceedings of the Seventh International Conference on Knowledge Discovery and Data Mining*, pages 204–213. ACM Press, 2001.

[16] B. Zadrozny, J. Langford, and N. Abe. Cost-sensitive learning by cost-proportionate example weighting. In *Proceedings of the Third IEEE International Conference on Data Mining*, pages 435–442, 2003.

[17] T. Zhang. Sequential greedy approximation for certain convex optimization problems. volume IT-49, pages 682–691, 2003.