Adversarial Cost-Sensitive Classification

Kaiser Asif **Wei Xing** **Sima Behpour** **Brian D. Ziebart**
Department of Computer Science
University of Illinois at Chicago
{kasif2,wxing3,sbehpo2,bziebart}@uic.edu

Abstract

In many classification settings, mistakes incur different application-dependent penalties based on the predicted and actual class labels. Cost-sensitive classifiers minimizing these penalties are needed. We propose a robust minimax approach for producing classifiers that directly minimize the cost of mistakes as a convex optimization problem. This is in contrast to previous methods that minimize the empirical risk using a convex surrogate for the cost of mistakes, since minimizing the empirical risk of the actual cost-sensitive loss is generally intractable. By treating properties of the training data as uncertain, our approach avoids these computational difficulties. We develop theory and algorithms for our approach and demonstrate its benefits on cost-sensitive classification tasks.

1 INTRODUCTION

In many applications of machine learning, the penalty or cost for classification errors depends on both the predicted label and the actual label. For example, an incorrect disease diagnosis may lead to treatments that cause complications of varying severity depending on the patient's actual disease. These different incurred penalties for mistakes can be represented as a confusion cost matrix that is indexed by the predicted class (row) and actual class (column). As shown in the following confusion cost matrix for a classification task with four possible labels,

$$\mathbf{C} = \begin{bmatrix} 0 & 1 & 2 & 0 \\ 3 & 0 & 1 & 3 \\ 4 & 2 & 0 & 1 \\ 1 & 1 & 2 & 0 \end{bmatrix}, \tag{1}$$

the confusion costs need not be symmetric or possess any other specific relationships. Here, correct predictions incur zero cost ($C_{i,i} = 0$), but even this property is not required. Additionally, other classification errors may incur zero cost ($C_{1,4} = 0$) if, e.g., the same treatment cures two different diseases. Note that the zero-one loss is a special case with off-diagonal values of one and on-diagonal costs of zero.

A natural goal for machine learning is to obtain a classifier that minimizes the expected cost incurred when classifying an example. Previous research primarily takes existing classification methods based on empirical risk minimization and tries to adapt them in various ways to be sensitive to these misclassification costs. Reweighting methods artificially augment the training data with copies of "high cost" examples to make the classifier more cost-sensitive to them [Chan and Stolfo, 1998, Elkan, 2001, Zadrozny et al., 2003, Zhou and Liu, 2010]. Other methods modify the criteria used to obtain a classifier that incorporates mistake-specific losses [Knoll et al., 1994, Turney, 1995, Elkan, 2001, Brefeld et al., 2003, Ling et al., 2004, Lomax and Vadera, 2013]. However, in both cases the non-convexity of the cost-sensitive loss function makes empirical risk minimization impractical [Hoffgen et al., 1995]. Surrogate loss functions that are convex (e.g., the hinge loss) are instead minimized, but this can introduce significant suboptimality.

Rather than integrating cost-sensitivity into existing machine learning techniques, we formulate a new machine learning approach from first principles to robustly minimize the expected cost. Our approach treats classifier construction as a game against an adversarial evaluator [Topsøe, 1979, Grünwald and Dawid, 2004]. This enables us to directly minimize the cost-sensitive loss on an approximation of the training data instead of using a convex approximation of the cost-sensitive loss, as is done with empirical risk minimization. Inference reduces to solving a zero-sum game in our approach. This is efficiently accomplished using linear programming. We obtain parameter estimates by constructing game payoff parameters using convex optimization methods. The key benefit of our approach is that the exact confusion cost matrix is employed rather than a convex surrogate. We provide important bounds on the generalization error and demonstrate the conceptual and empirical benefits of our approach in practice.

2 PRELIMINARIES & RELATED WORK

2.1 EMPIRICAL RISK MINIMIZATION

A standard approach to parametric classification is to assume some functional form for the classifier (e.g., a linear discriminant function, $f_\theta(\mathbf{x}) = \text{argmax}_y \, \theta^{\mathrm{T}} \phi(\mathbf{x}, y)$, where $\phi(\mathbf{x}, y) \in \mathbb{R}^k$ is a feature function) and then select model parameters θ that minimize the empirical risk,

$$\underset{\theta}{\text{argmin}} \, \mathbb{E}_{\tilde{P}(\mathbf{x},y)} \left[\text{loss} \left(Y, f_\theta(\mathbf{X}) \right) \right] + \lambda ||\theta||, \qquad (2)$$

with a regularization penalty $\lambda||\theta||$ often added to avoid overfitting to available training data[1]. Unfortunately, many combinations of classification functions, $f_\theta(\mathbf{x})$, and loss functions, $\text{loss}(\cdot, \cdot)$, do not lend themselves to efficient parameter optimization under the empirical risk minimization (ERM) formulation. For example, the zero-one loss measuring the misclassification rate will generally lead to a non-convex empirical risk minimization problem that is NP-hard to solve [Hoffgen et al., 1995].

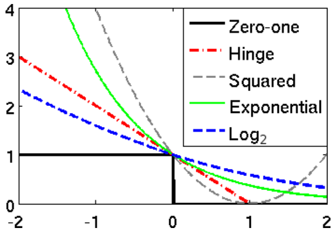

Figure 1: Convex surrogates for the zero-one loss.

To avoid these intractabilities, convex surrogate loss functions (Figure 1) that serve as upper bounds on the desired loss function are often used to create tractable optimization problems. The popular support vector machine (SVM) classifier [Cortes and Vapnik, 1995], for example, employs the hinge-loss—an upper bound on the zero-one loss—to avoid the often intractable empirical risk minimization problem. Adaboost [Freund and Schapire, 1997] incrementally minimizes the exponential loss. The difference between the actual loss and its convex surrogate can introduce a substantial mismatch between optimal parameter estimation under the surrogate loss function and optimal parameter estimates for the original performance objective.

2.2 COST-SENSITIVE LEARNING

Cost-sensitive learning considers more general loss functions than the zero-one loss in which the loss depends on the actual and the predicted class. One approach is to estimate the conditional label distribution, $\hat{P}(y|\mathbf{x})$, and employ the Bayesian optimal classifier: $\hat{f}(\mathbf{x}) = \text{argmin}_{y' \in \mathcal{Y}} \mathbb{E}_{\hat{P}(y|\mathbf{x})}[C_{y',Y}]$, using, e.g., the cost matrix of Eq. (1). However, accurately estimating the conditional label distribution will typically require much more data than methods that directly learn the best class prediction for a given loss function [Margineantu, 2002].

Early meta-learning methods for cost-sensitive learning attempt to modify how a cost-insensitive learner is used during training and/or prediction time so that the end result of its use is cost-sensitive. One approach for this is to either stratify or reweight available training data so that more costly mistakes will incur a larger overall cost and therefore the resulting classifier will be more sensitive to them [Chan and Stolfo, 1998, Elkan, 2001, Zadrozny et al., 2003, Zhou and Liu, 2010]. However, the validity of this approach is limited to a restricted class of *consistent* cost matrices when applied to multi-class prediction tasks [Domingos, 1999, Zhou and Liu, 2010]. A method that reduces multi-class predictions to binary predictions using iterative reweighting, data space expansion, and gradient boosting with stochastic ensembles [Abe et al., 2004] has been proposed to overcome these limitations. The *Metacost* algorithm [Domingos, 1999] similarly wraps around any underlying classifier. It uses bagging to produce label probability estimates, which it then uses to modify training data labels to produce more cost-sensitive predictions on the training set.

Direct cost-sensitive learning methods incorporate the confusion costs directly into the formulation of the classifier. Some classification methods are much more amenable to cost-sensitive modifications than others. In decision trees, for example, modified criteria for greedily selecting decision nodes and/or pruning the tree based on the confusion cost have been successfully employed [Knoll et al., 1994, Turney, 1995, Elkan, 2001, Ling et al., 2004, Davis et al., 2006, Lomax and Vadera, 2013], while relatively little attention has been given for developing cost-sensitive nearest neighbor classifiers [Qin et al., 2013].

Boosting iteratively creates an ensemble of weak classifiers that are then combined to create a much stronger classifier [Freund and Schapire, 1997] that often performs well in practice. Cost-sensitive boosting techniques employ cost-sensitive weak learners to produce a stronger learner that is cost-sensitive as well [Fan et al., 1999, Ting, 2000]. This is accomplished by minimizing the risk over the training dataset, $\frac{1}{n}\sum_{i=1}^{n} \text{loss}'(C, y_i, S(\mathbf{x}_i))$, using a generalized surrogate loss function, $\text{loss}'(C, \tilde{y}, S_m(\mathbf{x}))$, for the cost matrix C, class label $\tilde{y}$, and where $S_y(\mathbf{x})$ represents the classifier confidence in assigning class y to data point $\mathbf{x}$. Recently developed loss functions are the Generalized Exponential Loss (GEL), $\sum_{y'} C_{y,y'} e^{S_{y'}(\mathbf{x}) - S_y(\mathbf{x})}$ and the Generalized Logistic Loss (GLL), $\log(1 + \sum_{y'} C_{y,y'} e^{S_{y'}(\mathbf{x}) - S_y(\mathbf{x})})$. These loss functions are guess-averse and produce state-of-the-art perfor-

[1] Lowercase non-bold, x, and bold, $\mathbf{x}$, denote scalar and vector values, and capitals, X or $\mathbf{X}$, denote random variables.

mance when used in boosting for cost-sensitive classification [Beijbom et al., 2014].

Support vector machines [Cortes and Vapnik, 1995] have been generalized in the binary classification setting by penalizing mistakes for one class more than for the other class [Brefeld et al., 2003]. Multiclass problems are reduced to binary classifiers using one-versus-all [Bottou et al., 1994] and one-versus-one [Knerr et al., 1990] prediction tasks. The *Cost-Sensitive One-Versus-All* (CSOVA) algorithm [Lin, 2008] trains a separate binary SVM classifier for each class. The *Cost-Sensitive One-Versus-One* (CSOVO) algorithm [Lin, 2010] instead constructs a total of $k(k-1)/2$ classifiers—one for each pair of classes (i, j). For both CSOVA and CSOVO, binary classifiers are aggregated to produce a multi-class prediction. Using structured SVM methods [Tsochantaridis et al., 2005] to directly incorporate cost-sensitivity into the multiclass generalization of the hinge loss [Lee et al., 2004],

$$\min_{\theta,\, \epsilon \geq \mathbf{0}} \theta \cdot \theta + \alpha \sum_i \epsilon_i \text{ such that:} \tag{3}$$
$$\theta \cdot \phi(\mathbf{x}_i, y_i) - \theta \cdot \phi(\mathbf{x}_i, y') \geq C_{y', y_i} - \epsilon_i, \forall i, y' \neq y_i,$$

creates a margin-based classifier that incorporates mistake costs additively. We note that central to each of these SVM-based methods is the hinge loss approximation of the cost-sensitive loss function. Our approach avoids such approximations of the loss function by instead approximating the available training data.

2.3 ADVERSARIAL METHODS

The adversarial perspective that we leverage in our approach has played a formative role in statistical estimation and decision making under uncertainty. These include Wald's maximin model [Wald, 1949] of decision making as a sequential adversarial game, Savage's minimax optimization of the regret of decisions [Savage, 1951], and statistical estimates under uncertainty that minimize worst-case risk [Wolfowitz, 1950]. We follow a relaxation of this idea, which estimates complete probability distributions as solutions to a minimax game [Topsøe, 1979, Grünwald and Dawid, 2004]. This formulation is most commonly known as a means for deriving the principle of maximum entropy using the logarithmic loss. From this, many exponential family distributions (e.g., Gaussian distribution, exponential) can be derived [Wainwright and Jordan, 2008].

Our approach differs substantially from adversarial machine learning formulations that are made robust to adversarial shifts in the dataset [Dalvi et al., 2004, Liu and Ziebart, 2014] or uncertainty in the loss function [Wang and Tang, 2012]. We assume training and testing data are IID and the cost-sensitive loss function is fully known. We restrict our uncertainty to the conditional label distribution $P(y|\mathbf{x})$ and adversarially estimate it. In contrast with minimax approaches to classification that assume parametric forms of the data [Lanckriet et al., 2003], our approach allows the estimation of any conditional label distribution. Instead, only training data properties are incorporated in the form of constraints on the adversary's conditional label distribution [Grünwald and Dawid, 2004].

3 ADVERSARIAL COST-SENSITIVITY

3.1 FORMULATION

We begin to define our notation by considering an estimator for the conditional label distribution, $\hat{P}(y|\mathbf{x})$, the actual evaluation distribution $P(y|\mathbf{x})$, and an adversarial distribtion $\check{P}(y|\mathbf{x})$. We compactly represent each as $|\mathcal{Y}|$-sized vectors $\hat{\mathbf{p}}_\mathbf{x} = [\hat{P}(\hat{Y} = 1|\mathbf{x}) \ \hat{P}(\hat{Y} = 2|\mathbf{x}) \ \ldots]^\mathrm{T}$ for each input $\mathbf{x} \in \boldsymbol{\mathcal{X}}$, and, similarly, $\mathbf{p}_\mathbf{x}$ and $\check{\mathbf{p}}_\mathbf{x}$. The expected loss suffered from this estimator on input x for a confusion cost matrix $\mathbf{C}$ is: $\hat{\mathbf{p}}_\mathbf{x}^\mathrm{T} \mathbf{C} \mathbf{p}_\mathbf{x} = \mathbb{E}_{\hat{P}(\hat{y}|\mathbf{x})P(y|\mathbf{x})}[C_{\hat{Y},Y}]$.

Only samples from the true conditional label distribution $P(y|\mathbf{x})$ are available. We denote these by distribution $\tilde{P}(y|\mathbf{x})$ (compactly represented as $\tilde{\mathbf{p}}_\mathbf{x}$) and also input sample distribution $\tilde{P}(\mathbf{x})$. Minimizing the empirical risk under this distribution, $\mathbb{E}_{\tilde{P}(\mathbf{x})}[\hat{\mathbf{p}}_{\theta,\mathbf{X}}^\mathrm{T} \mathbf{C} \tilde{\mathbf{p}}_\mathbf{X}] = \frac{1}{n} \sum_{i=1}^n \sum_{\hat{y} \in \mathcal{Y}} \hat{P}(\hat{y}|\mathbf{x}_i) C_{\hat{y}, y_i}$, for some parametric form of the estimation distribution, e.g., $\hat{P}_\theta(y|\mathbf{x}) \propto e^{\theta \cdot \phi(\mathbf{x}, y)}$, leads to a non-convex and generally intractable optimization problem, assuming $\mathbf{P} \neq \mathbf{NP}$, as discussed in §2.1.

To avoid these non-convex optimization concerns, we employ a robust minimax formulation [Topsøe, 1979, Grünwald and Dawid, 2004] to construct our cost-sensitive classifier (Definition 1). This formulation views the estimation task as a two-player game between estimator and adversary. The adversary is constrained to choose distributions that match a vector of moment statistics of the distribution, $\mathbb{E}_{P(\mathbf{x})P(y|\mathbf{x})}[\phi(\mathbf{X}, Y)]$. We denote the set of conditional distributions $P(y|\mathbf{x})$ satisfying these statistics as Ξ.

Definition 1. *In the* **constrained cost-sensitive minimax game**, *the estimator player first selects a predictive distribution, $\hat{\mathbf{p}}_\mathbf{x} \triangleq \hat{P}(\hat{y}|\mathbf{x}) \in \Delta$, for each input $\mathbf{x}$, from the conditional probability simplex Δ, and then the adversarial player selects an evaluation distribution, $\check{\mathbf{p}}_\mathbf{x} \triangleq \check{P}(\check{y}|\mathbf{x}) \in \Delta$, for each input x from the set Ξ of distributions consistent with known statistics:*

$$\min_{\{\hat{\mathbf{p}}_\mathbf{x}\} \in \boldsymbol{\Delta}} \ \max_{\{\check{\mathbf{p}}_\mathbf{x}\} \in \Xi \cap \boldsymbol{\Delta}} \mathbb{E}_{P(\mathbf{x})}[\hat{\mathbf{p}}_\mathbf{X}^\mathrm{T} \mathbf{C} \check{\mathbf{p}}_\mathbf{X}] \tag{4}$$
$$\textit{where: } \Xi : \mathbb{E}_{P(\mathbf{x})\check{P}(\check{y}|\mathbf{x})}[\phi(\mathbf{X}, \check{Y})] = \tilde{\phi}.$$

We denote the set of conditional probabilities for each input $\mathbf{x}$ as $\{\hat{\mathbf{p}}_\mathbf{x}\}$ and $\{\check{\mathbf{p}}_\mathbf{x}\}$. Here, $\tilde{\phi}$ is a vector of provided feature moments measured from sample training data, $\tilde{\phi} = \mathbb{E}_{\tilde{P}(\mathbf{x},y)}[\phi(\mathbf{X}, Y)]$, for example.

Conceptually, the feature statistics $\phi(\mathbf{x}, y)$ defining the set Ξ should be chosen to restrict the adversary as much as possible from maximizing the loss. However, defining the set to be too restrictive leads to overfitting to the training data. Indeed, the complexity of the estimator $\hat{P}(\hat{y}|\mathbf{x})$ implicitly grows with the dimensionality of the constraints in Ξ. Thoughtfully specifying the feature function $\phi(\cdot,\cdot)$ and employing regularization can avoid this issue (§3.4).

3.2 INFERENCE AS ZERO-SUM GAME EQUILIBRIA

We establish efficient inference algorithms for our approach in this section. Theorem 1 transforms the joint adversary-constrained zero-sum games over many different inputs $\mathbf{x}$ into a set of unconstrained zero-sum game that are independent for each input $\mathbf{x}$ and connected by a parameterized cost matrix defining each player's game outcomes.

Theorem 1. *Determining the value of the constrained cost-sensitive minimax game reduces to a minimization over the expectation of many unconstrained minimax game:*

$$\min_{\{\hat{\mathbf{p}}_{\mathbf{x}}\}\in\boldsymbol{\Delta}} \max_{\{\check{\mathbf{p}}_{\mathbf{x}}\}\in\Xi\cap\boldsymbol{\Delta}} \mathbb{E}_{P(\mathbf{x})}[\hat{\mathbf{p}}_{\mathbf{X}}^{\mathrm{T}}\mathbf{C}\check{\mathbf{p}}_{\mathbf{X}}] \qquad (5)$$

$$= \max_{\{\check{\mathbf{p}}_{\mathbf{x}}\}\in\Xi\cap\boldsymbol{\Delta}} \mathbb{E}_{P(\mathbf{x})}\left[\min_{\hat{\mathbf{p}}_{\mathbf{X}}\in\Delta} \hat{\mathbf{p}}_{\mathbf{X}}^{\mathrm{T}}\mathbf{C}\check{\mathbf{p}}_{\mathbf{X}}\right] \qquad (6)$$

$$= \min_{\theta} \mathbb{E}_{P(\mathbf{x})}\left[\max_{\check{\mathbf{p}}_{\mathbf{X}}\in\Delta} \min_{\hat{\mathbf{p}}_{\mathbf{X}}\in\Delta} \hat{\mathbf{p}}_{\mathbf{X}}^{\mathrm{T}}\mathbf{C}'_{\mathbf{X},\theta}\check{\mathbf{p}}_{\mathbf{X}}\right], \qquad (7)$$

where θ parametrizes the new game characterized by matrix $\mathbf{C}'_{\mathbf{x},\theta} : (C'_{\mathbf{x},\theta})_{\hat{y},\check{y}} = C_{\hat{y},\check{y}} + \theta^{\mathrm{T}}(\phi(\mathbf{x},\check{y}) - \phi(\mathbf{x},\tilde{y}))$, and $\phi(\cdot,\cdot)$ terms are from the definition of set Ξ.

Proof of Theorem 1.

$$\min_{\{\hat{\mathbf{p}}_{\mathbf{x}}\}\in\boldsymbol{\Delta}} \max_{\{\check{\mathbf{p}}_{\mathbf{x}}\}\in\Xi\cap\boldsymbol{\Delta}} \mathbb{E}_{P(\mathbf{x})}[\hat{\mathbf{p}}_{\mathbf{X}}^{\mathrm{T}}\mathbf{C}\check{\mathbf{p}}_{\mathbf{X}}]$$

$$\overset{(a)}{=} \max_{\{\check{\mathbf{p}}_{\mathbf{x}}\}\in\Xi\cap\boldsymbol{\Delta}} \min_{\{\hat{\mathbf{p}}_{\mathbf{x}}\}\in\boldsymbol{\Delta}} \mathbb{E}_{P(x)}[\hat{\mathbf{p}}_{\mathbf{X}}^{\mathrm{T}}\mathbf{C}\check{\mathbf{p}}_{\mathbf{X}}]$$

$$\overset{(b)}{=} \max_{\{\check{\mathbf{p}}_{\mathbf{x}}\}\in\Xi\cap\boldsymbol{\Delta}} \mathbb{E}_{P(\mathbf{x})}\left[\min_{\hat{\mathbf{p}}_{\mathbf{X}}\in\Delta} \hat{\mathbf{p}}_{\mathbf{X}}^{\mathrm{T}}\mathbf{C}\check{\mathbf{p}}_{\mathbf{X}}\right]$$

$$\overset{(c)}{=} \max_{\{\check{\mathbf{p}}_{\mathbf{x}}\}\in\boldsymbol{\Delta}} \min_{\theta} \mathbb{E}_{P(\mathbf{x})}\left[\min_{\hat{\mathbf{p}}_{\mathbf{X}}\in\Delta} \hat{\mathbf{p}}_{\mathbf{X}}^{\mathrm{T}}\mathbf{C}\check{\mathbf{p}}_{\mathbf{X}}\right] + \theta^{\mathrm{T}}\mathbb{E}_{P(\mathbf{x})}[\boldsymbol{\Phi}_{\mathbf{X}}(\check{\mathbf{p}}_{\mathbf{X}} - \tilde{\mathbf{p}}_{\mathbf{X}})]$$

$$\overset{(d)}{=} \min_{\theta} \mathbb{E}_{P(\mathbf{x})}\left[\max_{\check{\mathbf{p}}_{\mathbf{X}}\in\Delta} \min_{\hat{\mathbf{p}}_{\mathbf{X}}\in\Delta} \hat{\mathbf{p}}_{\mathbf{X}}^{\mathrm{T}}\mathbf{C}'_{\mathbf{X},\theta}\check{\mathbf{p}}_{\mathbf{X}}\right]$$

where $\boldsymbol{\Phi}$ is the matrix defined by $\boldsymbol{\Phi}_{i,j} = \phi_i(\mathbf{x}, y_j)$ and $\mathbf{C}'_{\mathbf{x}}$ is defined by elements:

$$(C'_{\mathbf{x}})_{\hat{y},\check{y}} = C_{\hat{y},\check{y}} + \theta^{\mathrm{T}}(\phi(\mathbf{x},\check{y}) - \phi(\mathbf{x},\tilde{y})). \qquad (8)$$

Step (a) follows from minimax duality in zero-sum games [von Neumann and Morgenstern, 1947]. As an affine function of terms each with individual $\check{\mathbf{p}}_{\mathbf{x}}$ term, each minimization can be performed independently in step (b). Step (c) expresses the primal Lagrangian. For step (d), $\mathbb{E}_{P(\mathbf{x})}[\min_{\hat{\mathbf{p}}_{\mathbf{X}}\in\Delta} \hat{\mathbf{p}}_{\mathbf{X}}^{\mathrm{T}}\mathbf{C}\check{\mathbf{p}}_{\mathbf{X}} + \theta^{\mathrm{T}}\boldsymbol{\Phi}_{\mathbf{X}}(\check{\mathbf{p}}_{\mathbf{X}} - \tilde{\mathbf{p}}_{\mathbf{X}})]$—a non-negative linear combination of minimums of affine functions—is a concave function of $\check{\mathbf{p}}_{\mathbf{x}}$ terms. Given a feasible solution on the relative interior of Ξ [Boyd and Vandenberghe, 2004], strong Lagrangian duality holds. As in step (b), the maximizations can then be independently applied. □

Figure 2 shows the value of the game for a single $\mathbf{x}$ from Eq. (6) as a function of the adversial distribution $\check{\mathbf{p}}_{\mathbf{x}}$ for zero-one loss and a more general cost matrix. The adversary is not free to independently maximize these functions for each $\mathbf{x}$, but must instead choose a structured prediction that resides within the constraint set Ξ.

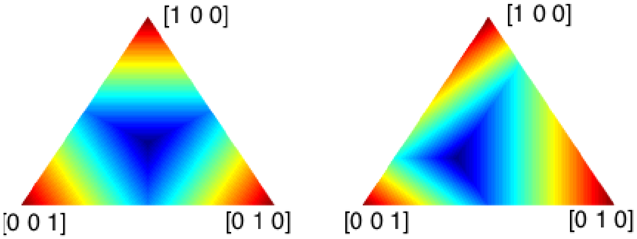

Figure 2: The portion of the adversary's objective function (6) for a single example, $\min_{\hat{\mathbf{p}}_{\mathbf{x}}\in\Delta} \hat{\mathbf{p}}_{\mathbf{x}}^{\mathrm{T}}\mathbf{C}\check{\mathbf{p}}_{\mathbf{x}}$, in the adversary-constrained game for zero-one loss (left) and a more general cost-sensitive loss with cost matrix [0 2 3; 2 0 1; 1 3 0] (right) in a three-class prediction task.

After applying Theorem 1 and given model parameters, θ, (obtaining these parameters is discussed in §3.3) the unconstrained game, $\max_{\check{\mathbf{p}}_{\mathbf{x}}\in\Delta} \min_{\hat{\mathbf{p}}_{\mathbf{x}}\in\Delta} \hat{\mathbf{p}}_{\mathbf{x}}^{\mathrm{T}}\mathbf{C}'_{\mathbf{x},\theta}\check{\mathbf{p}}_{\mathbf{x}}$, can be solved independently for each $\mathbf{x}$. In this augmented game, our original cost matrix from Eq. (1) is transformed into the augmented cost matrix:

$$\mathbf{C}' = \begin{bmatrix} 0+\psi_1 & 1+\psi_2 & 2+\psi_3 & 0+\psi_4 \\ 3+\psi_1 & 0+\psi_2 & 1+\psi_3 & 3+\psi_4 \\ 4+\psi_1 & 2+\psi_2 & 0+\psi_3 & 1+\psi_4 \\ 1+\psi_1 & 1+\psi_2 & 2+\psi_3 & 0+\psi_4 \end{bmatrix}, \qquad (9)$$

where Lagrangian potentials are compactly denoted as $\psi_i = \theta^{\mathrm{T}}(\phi(\mathbf{x}, i) - \phi(\mathbf{x}, \tilde{y}))$ with $\tilde{y}$ representing the example's actual label. For parameter estimation, the second feature function based on the actual label $\tilde{y}$ serves an important role. However, since it is constant with respect to $\check{y}$ and $\hat{y}$, and therefore does not influence the solution strategies for the game, it can be ignored when making predictions on data with unknown labels (or assigned an arbitrary value from $\mathcal{Y}$ without affecting predictions).

Figure 3 shows the adversary's objective function in the unconstrained, cost-augmented game. Conceptually, the adversary's objective function from the constrained game

(Figure 2) is "placed" on top of a hyperplane shaped by the Lagrangian potential terms, ψ_i. The difference in these potential terms determines the adversary's equilibrium strategy. For the binary classification task, there are three possible equilibrium strategies for the adversary. With three classes, there are seven possibilities: three pure strategies; three strategies that are mixtures of two classes; and one strategy that is a mixture of all three classes.

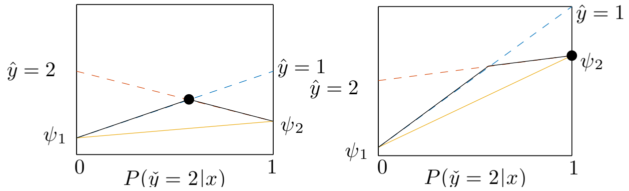

Figure 3: The adversary's objective in the unconstrained game for a binary classification task with a mixed (uncertain) equilibrium solution (left) and a pure (certain) equilibrium solution (right). The third adversary strategy, $P(\check{y}=2|\mathbf{x})=0$, is realized when $\psi_1 >> \psi_2$.

Unlike the logarithmic loss under this minimax formulation, the cost-sensitive loss function does not provide a closed-form parametric solution[2]. Instead, the inner minimax game (inside the expectation of Eq. (7)) for each input $\mathbf{x}$ can be solved as a linear program [von Neumann and Morgenstern, 1947]:

$$\max_{v,\check{P}(\check{y}|\mathbf{x})} \; v \tag{10}$$
$$\text{subject to: } v \leq \sum_{\check{y}\in\mathcal{Y}} \check{P}(\check{y}|\mathbf{x})(C'_{\mathbf{x},\theta})_{\hat{y},\check{y}} \; \forall\hat{y}\in\mathcal{Y}$$
$$\sum_{\check{y}\in\mathcal{Y}} \check{P}(\check{y}|\mathbf{x}) = 1 \text{ and } \check{P}(\check{y}|\mathbf{x}) \geq 0, \; \forall\check{y}\in\mathcal{Y}.$$

The resulting distribution, $\check{P}(\check{y}|\mathbf{x})$, gives the adversary's strategy $\check{\mathbf{p}}^*_{\mathbf{x}}$. The other strategy of the Nash equilibrium strategy pair, $(\check{\mathbf{p}}^*_{\mathbf{x}}, \hat{\mathbf{p}}^*_{\mathbf{x}})$ can be obtained by solving the same linear program with the cost matrix transposed and negated.

3.3 LEARNING VIA CONVEX OPTIMIZATION

Our key remaining task for employing the proposed approach is to obtain model parameters (Lagrangian multipliers) θ that enforce the adversarial distribution to reside within the constraint set Ξ.

Theorem 2. *The subdifferential of the outer minimization problem (Eq.* (7)*) includes the expected feature difference as a subgradient:*

$$\mathbb{E}_{P(\mathbf{x})\check{P}^*_{\hat{\theta}}(\check{y}|\mathbf{x})}\left[\phi(\mathbf{X},\check{Y})\right] - \mathbb{E}_{P(\mathbf{x})P(y|\mathbf{x})}\left[\phi(\mathbf{X},Y)\right] \tag{11}$$
$$\in \partial_\theta \mathbb{E}_{P(\mathbf{x})}\left[\min_{\hat{\mathbf{p}}_\mathbf{x}\in\Delta}\max_{\check{\mathbf{p}}_\mathbf{x}\in\Delta} \hat{\mathbf{p}}_\mathbf{X}^{\mathrm{T}}\mathbf{C}'_{\mathbf{X},\theta}\check{\mathbf{p}}_\mathbf{X}\right]\Bigg|_{\theta=\hat{\theta}}$$

where $\check{P}^*(\check{y}|\mathbf{x})$ is the solution to Eq. (10).

Proof of Theorem 2. Taking the subdifferential, we have:

$$\partial_{\theta_k}\mathbb{E}_{P(\mathbf{x})}\left[\min_{\hat{\mathbf{p}}_\mathbf{x}\in\Delta}\max_{\check{\mathbf{p}}_\mathbf{x}\in\Delta} \hat{\mathbf{p}}_\mathbf{X}^{\mathrm{T}}\mathbf{C}'_{\mathbf{X},\theta}\check{\mathbf{p}}_\mathbf{X}\right]\Bigg|_{\theta=\hat{\theta}}$$
$$\overset{(a)}{=} \mathbb{E}_{P(\mathbf{x})}\left[\partial_{\theta_k}\max_{\check{\mathbf{p}}_\mathbf{x}\in\Delta}\min_{\hat{\mathbf{p}}_\mathbf{x}\in\Delta} \hat{\mathbf{p}}_\mathbf{X}^{\mathrm{T}}\mathbf{C}'_{\mathbf{X},\theta}\check{\mathbf{p}}_\mathbf{X}\right]\Bigg|_{\theta=\hat{\theta}}$$
$$\overset{(b)}{\ni} \mathbb{E}_{P(\mathbf{x})}\left[\partial_{\theta_k}\left(\hat{\mathbf{p}}^*_\mathbf{X}\right)^{\mathrm{T}}\mathbf{C}'_{\mathbf{X},\theta}\check{\mathbf{p}}^*_\mathbf{X}\right]\Big|_{\theta=\hat{\theta}}$$
$$\overset{(c)}{=} \mathbb{E}_{P(\mathbf{x})}\left[\left(\hat{\mathbf{p}}^*_\mathbf{X}\right)^{\mathrm{T}}\left(\partial_{\theta_k}\mathbf{C}'_{\mathbf{X},\theta}\right)\check{\mathbf{p}}^*_\mathbf{X}\right]\Big|_{\theta=\hat{\theta}}$$
$$\overset{(d)}{\ni} \mathbb{E}_{P(\mathbf{x})\check{P}^*_{\hat{\theta}}(\check{y}|\mathbf{x})}\left[\phi_k(\mathbf{X},\check{Y})\right] - \mathbb{E}_{P(\mathbf{x})P(y|\mathbf{x})}\left[\phi_k(\mathbf{X},Y)\right].$$

Step (a) follows from the rule for non-negative combinations of subdifferentials. Step (b) follows from the subdifferential of the function evaluated at the maximizing/minimizing values being a subset of the subdifferential of the maximum/minimum functions. Step (c), like step (a), follows from the rule for non-negative combinations of subdifferentials by noting that $\left(\hat{\mathbf{p}}^*_\mathbf{X}\right)^{\mathrm{T}}\mathbf{C}'_{\mathbf{X},\theta}\check{\mathbf{p}}^*_\mathbf{X} = \hat{\mathbf{p}}^*_\mathbf{X}\left(\check{\mathbf{p}}^*_\mathbf{X}\right)^{\mathrm{T}} \bullet \mathbf{C}'_{\mathbf{X},\theta}$, where $\bullet$ represents the "matrix dot product" (i.e., $\mathbf{A}\bullet\mathbf{B} \triangleq \sum_{i,j} A_{i,j}B_{i,j}$). In step (d), the subdifferential terms for $\mathbf{C}'_\mathbf{x}$ include $\phi_k(\mathbf{x},\check{y}) - \phi_k(\mathbf{x},\tilde{y}) \in (\partial_{\theta_k}\mathbf{C}'_\mathbf{x})_{\hat{y},\check{y}}$ and do not depend on $\hat{\mathbf{p}}_\mathbf{x}$. □

Leveraging the convexity of the formulation's objective function (discussed in the Proof of Theorem 1), and using the common substitution of the sample training data distribution, $\tilde{P}(\mathbf{x})$, in place of the distribution $P(\mathbf{x})$, we employ standard subgradient-based optimization methods for convex optimization problems to obtain parameters for our cost-sensitive classifier (Algorithm 1).

Algorithm 1 Parameter estimation for the robust cost-sensitive classifier

```
Input: Cost matrix $\mathbf{C}$, training dataset $\mathcal{D}$ with pairs
    $(\tilde{\mathbf{x}}_i, \tilde{y}_i) \in \mathcal{D}$, feature function $\phi : \boldsymbol{\mathcal{X}} \times \mathcal{Y} \rightarrow \mathbb{R}^k$, time-
    varying learning rate $\{\gamma_t\}$
Output: Model parameter estimate $\theta$
  $t \leftarrow 1$
  while $\theta$ not converged do
    for all $(\tilde{\mathbf{x}}_i, \tilde{y}_i) \in \mathcal{D}$ do
      Construct cost matrix $\mathbf{C}'_{\tilde{\mathbf{x}}_i,\theta}$ using Eq. (8)
      Solve for $\check{P}(\check{y}|\tilde{\mathbf{x}}_i)$ using the LP of Eq. (10)
      $\nabla_\theta = \mathbb{E}_{\check{P}(\check{y}|\tilde{\mathbf{x}}_i)}[\phi(\tilde{\mathbf{x}}_i, \check{Y})] - \phi(\tilde{\mathbf{x}}_i, \tilde{y}_i)$
      $\theta = \theta - \gamma_t \nabla_\theta$
      $t \leftarrow t + 1$
    end for
  end while
```

Though we describe a stochastic subgradient in our algorithm, any convex optimization method for non-smooth objective functions can be employed.

[2] Adversarial logarithmic loss minimization yields members of the exponential family [Wainwright and Jordan, 2008].

3.4 PERFORMANCE GUARANTEES & ILLUSTRATIVE EXAMPLES

We establish performance guarantees and illustrate the behavior of our approach in this portion of the paper. We focus specifically on the similarities to and differences from support vector machines [Cortes and Vapnik, 1995] and their structured extensions [Tsochantaridis et al., 2004]. Given ideal data (linearly separable), Theorem 3 establishes an equivalence to hard-margin SVMs.

Theorem 3. *Given linearly separable training data, i.e.,*

$$\exists\theta : \forall i, y' \neq y_i, \theta \cdot \phi(\mathbf{x}_i, y_i) > \theta \cdot \phi(\mathbf{x}_i, y'), \qquad (12)$$

and zero cost only for correct predictions $C_{i,i} = 0$, the adversarial cost-sensitive learner with sufficiently small L_2 regularization is equivalent to a hard-margin cost-sensitive support vector machine.

Proof. Eq. (12) implies $\exists\theta' : \forall i, y' \neq y_i, \theta' \cdot \phi(\mathbf{x}_i, y_i) > \theta' \cdot \phi(\mathbf{x}_i, y') + C_{y',y_i}$ (the hard-margin cost-sensitive SVM constraint set with $\epsilon = \mathbf{0}$ in Eq. (3)) by multiplicatively scaling θ. The Nash equilibrium is $\check{P}(\check{y}_i|\mathbf{x}_i) = 1$ and $\hat{P}(\hat{y}_i|\mathbf{x}_i) = 1$ with a cost-sensitive loss of zero *if and only if* this inequality is satisfied. Given this, the dual optimization in Eq. (7) realizes its minima (zero loss) only when these constraints are satisfied. The L_2 regularization term is a monotonic transformation of the objective of the hard-margin SVM: $\theta \cdot \theta$. Thus, having the same constraints and objective functions with corresponding maxima, an equivalent solution is produced. □

As a result of this equivalence to hard-margin SVM, adversarial classification inherits the convergence properties of support vectors machines in the realizable case of Eq. (12).

The game strategies of each player are illustrated in Figure 4 for binary prediction using the zero-one loss in the separable setting. Between perfectly classified datapoints, our approach produces a region of uncertainty that is maximally uncertain for the adversary's Nash equilibrium strategy ($\check{P}(\check{Y} = \text{‘o’}|\mathbf{x}) = 0.5$), while the predictor's Nash equilibrium strategy smoothly transitions from one class to the other in this region.

Given non-separable data, the adversarial approach suggests choosing a set Ξ of constraints based on training samples $\tilde{P}(\mathbf{x}, y)$ that will also contain the true label distribution, $P(y|\mathbf{x})$. When this is accomplished, Theorem 4 provides performance guarantees for generalization.

Theorem 4. *If $P(y|\mathbf{x}) \in \Xi$, confusion costs from the adversarial game upper bound the generalization error confusion costs:*

$$\mathbb{E}_{P(\mathbf{x})P(y|\mathbf{x})\hat{P}^*(\hat{y}|\mathbf{x})}[C_{\hat{Y},Y}] \leq \mathbb{E}_{P(\mathbf{x})\check{P}^*(\check{y}|\mathbf{x})\hat{P}^*(\hat{y}|\mathbf{x})}[C_{\hat{Y},\check{Y}}].$$

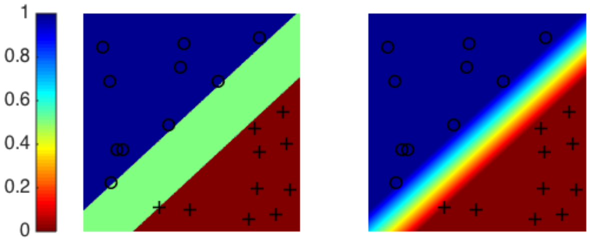

Figure 4: Adversary (left) and predictor (right) distributions for separable data under zero-one loss

Proof. By definition, the adversarial conditional label distribution, $\check{P}^*(\check{y}|\mathbf{x})$, is a Nash equilibrium and it provides the worst possible loss for the estimator of all conditional label distributions from set Ξ. So long as the true label distribution used for evaluation, $P(y|\mathbf{x})$, is similar to training data properties (i.e, a member of Ξ), then costs that are no worse than $\check{P}^*(\check{y}|\mathbf{x})$ can result without $P(y|\mathbf{x})$ being a better choice from Ξ than $P(y|\mathbf{x})$ for maximizing the predictor's loss, a contradiction. □

Slack can be added to the constraint set Ξ or regularization to the dual optimization problem of Eq. (6) to address finite sample approximation error when using sample data, $\mathbb{E}_{\tilde{P}(\mathbf{x},y)}[\phi(\mathbf{X}, Y)]$, as an estimate of the distribution's statistics, $\mathbb{E}_{P(\mathbf{x},y)}[\phi(\mathbf{X}, Y)]$.

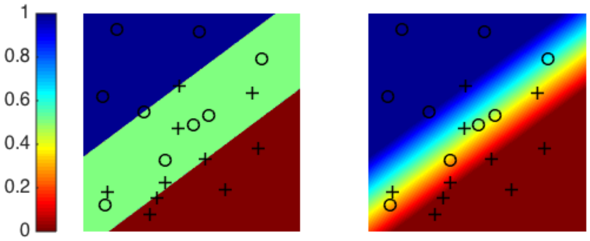

Figure 5: Adversary (left) and predictor (right) distributions for nonseparable data under zero-one loss

Figure 5 shows the two equilibria strategies for data that is not linearly separable in the zero-one loss binary classification setting. The uncertainty region of our approach depends on summary statistics rather than the specific datapoint labels that define margin boundaries of SVMs (Figure 5). Increased non-separability of the data and greater regularization amounts expand this uncertainty region.

The equilibria under cost-sensitive losses, shown in Figure 6 shifts the region of uncertainty to better minimize the expected cost compared to Figure 5, which is based on the same data sample. Additionally, the adversary's predictions shift ($\check{P}(Y = \text{‘o’}|\mathbf{x}) = .25$) within the region of uncertainty.

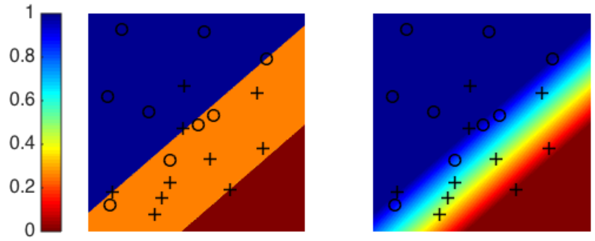

Figure 6: Adversary (left) and predictor (right) distributions for nonseparable data under [0 1; 3 0] cost matrix.

From the perspective of Theorem 3 and Theorem 4, adversarial cost-sensitive classification provides an alternative to hinge-loss "softening" of the hard-margin SVM. By posing cost-sensitive prediction as an adversarial game (Def. 1), our approach approximates aspects of the training data while being able to employ non-convex loss functions without the intractability encountered by empirical risk minimization. Prediction under this approach reduces to the well-studied problem of solving a zero-sum game, which is easily addressed using linear programming via Eq. (10). This is only a little more complicated than predictions for SVM based on the label that maximizes a linear potential function. Like SVMs, estimating model parameters can be posed as a convex optimization problem and solved using subgradient optimization methods (Alg. 1) under our approach.

4 EXPERIMENTS

Our adversarial approach provides the advantage of operating efficiently on non-convex cost-sensitive loss functions, but only through approximating the training data label information rather than minimizing loss on the actual labeled training data. We experimentally investigate the trade-off our approach provides in this section.

4.1 DATASETS

We employ publicly available datasets for multiclass classification to evaluate our approach. The number of classes and the number of examples (size) of each dataset are listed in Table 1. We rescale the attributes to [0,1] and enumerate the class labels.

4.2 METHODOLOGY

We conduct 10 cost-sensitive classification tasks for each dataset. We generate confusion cost matrices, $\mathbf{C}$, for each task by: (1) assigning all correct classifications a cost of zero ($C_{i,i} = 0, \forall i$); and (2) sampling the remaining elements of the cost matrix from the uniform distribution ($C_{i,j} \sim U[0,1], \forall i \neq j$). For each classification task, we split the data into training and testing sets as described in Table 1. We measure the expected cost of each method averaged over each of the 10 tasks.

Table 1: Evaluation datasets and dataset characteristics.

Name	**Classes**	**Attributes**	**Training**	**Testing**
Iris	3	4	120	30
Optical Digits	10	64	3823	1797
Satellite Image	6	36	4435	2000
Shuttle	7	9	43500	14500
Vehicle	4	18	658	188
Wine	3	4	142	36
Breast Tissue	6	9	85	21
Ecoli	8	7	269	67
Glass	6	9	171	43
Image Segment	7	19	210	2100
Libras	15	90	288	72
Pen Digits	10	16	7494	3498
Vertebral	3	6	248	62

4.3 COMPARISON METHODS

Our primary points of comparison for investigating this paper's central hypothesis—that adversarial data approximation produces better cost-sensitive classifiers than convex loss approximation—are support vector methods. However, we also compare with recently reported state-of-the-art cost-sensitive boosting methods. We implement and compare our proposed approach against the following specific methods for cost-sensitive learning. The methodological details for each approach are:

- **Our approach:** We train our method via Algorithm 1 using a quadratic expansion of the original attributes and a "one-hot" encoding of the class label, $\phi(\mathbf{x}, y) = [\text{vector}(\mathbf{x}\mathbf{x}^{\text{T}})I(y = 1); \text{vector}(\mathbf{x}\mathbf{x}^{\text{T}})I(y = 2); \ldots]$. To produce deterministic predictions, we "round" the estimator's Nash equilibrium strategy, $\hat{P}^*(\hat{y}|\mathbf{x})$ to the most probable label. This avoids the ambiguity of other methods for making deterministic predictions from mixed strategies (e.g., two or more actions may be the best response to the adversary's Nash equilibrium strategy).

- **Guess Averse Cost-Sensitive Boosting**: We employ the guess averse cost-sensitive boosting method and implementation [Beijbom et al., 2014] with GLL loss described in §2.1. (We also investigated GEL, but found it to be consistently and significantly outperformed by GLL.) We use a linear regression model as the weak learner.

- **Cost-Sensitive One-Versus-One (CSOVO)**: We employ the LIBSVM [Chang and Lin, 2011] implementation of the CSOVO SVM approach [Lin, 2010] described in §2.1.Our experiments use quadratic kernels

Table 2: CSOVO and CSOVA kernel parameters chosen using five-fold cross validation on the training set from $\gamma_1 \in \{0.125, 1, 2, 5, 10, 1/\text{number of features}\}$ and $\gamma_0 \in \{1, 2, 5, 10, 50, 100, 200, 300, ..., 900\}$.

	CSOVO		**CSOVA**	
Name	γ_1	γ_0	γ_1	γ_0
Iris	5	2	1	700
Optical Digits	1	2	5	2
Satellite Image	10	50	1	1
Shuttle	0.125	900	0.125	900
Vehicle	10	5	10	10
Wine	1	500	1	5
Breast Tissue	0.125	900	10	400
Ecoli	5	500	0.125	800
Glass	5	400	10	700
Image Segment	0.125	300	0.125	600
Libras	1	5	1	2
Pen Digits	0.125	700	5	5
Vertebral	0.125	600	0.125	500

[Chang and Lin, 2011], $K(u, v) = (\gamma_1 u'v + \gamma_0)^2$ to match the expressiveness of our approach. We run five-fold cross validation on the training set of every dataset to choose quadratic kernel parameters (shown in Table 2), and then we use these best parameters to train from the training set and construct the final classifier model[3] Finally, we evaluate the CSOVO performance by measuring the prediction cost on the test data.

- **Cost-Sensitive One-Versus-All (CSOVA)**: We similarly employ the LIBSVM implementation of the CSOVA SVM approach described in §2.1. Our methodology matches that of CSOVO for cross-validation (parameters shown in Table 2), training, and testing.

- **Structured SVM** (SVM-Struct): We employ the Large Scale Structured SVM (SVM LS) software package [Branson et al., 2013] to obtain a multiclass cost-sensitive predictor based on the additive cost-sensitive hinge loss of Eq. (3). SVM LS applies online subgradient methods [Ratliff et al., 2007] and sequential order optimization [Shalev-Shwartz et al., 2011] to improve efficiency. We evaluate the Online Dual Ascent (ODA) algorithm [Branson et al., 2013] as well as the Stochastic Gradient Descent (SGD) method for the purpose of our cost-sensitive experiments. We employ a trade-off parameter α of 100.

4.4 RESULTS

Figure 7 shows the average loss incurred by each approach on the 13 different datasets. Our method generally performs well on all of the datasets except *wine* and *libras* datasets and has a similar performance with boosting. SVM methods except SVM-CSOVO are strong on some of the datasets (*optdigits*, *pendigits*, *wine* and *libras*). For many datasets, the performance of the reduction-based SVM approaches is significantly worse than our approach and boosting and the multi-class structured SVM approach. The multi-class structured SVM approach specifically is significantly worse than our method on many of the datasets (*satimage*, *shuttle*, *vehicle*, *breast tissue*, *pendigits*, and *vertebral*), while only significantly better on the *optdigits* dataset.

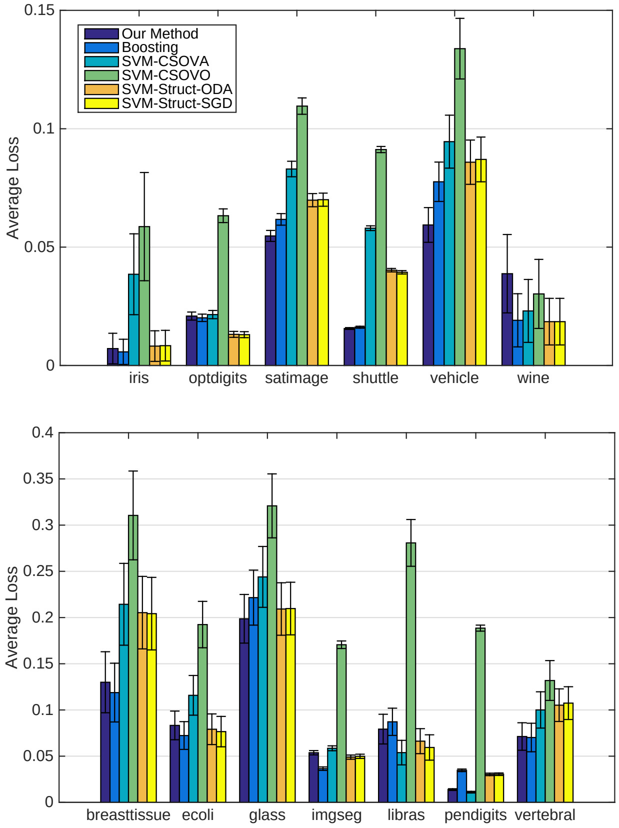

Figure 7: The average loss of predictions for the datasets of Table 1.

The differences between the results of our method and those of boosting are not as extreme. Indeed, for many of the datasets (*iris*, *wine*, *shuttle*, *optdigits*, *vertebral*, *ecoli*, *breast tissue*, and *libras*), the differences in average performance are not significant. For one dataset (*imgseg*), boosting is significantly better, while our method is significantly better for the remaining four (*satimage*, *shuttle*, *vehicle*, and *pendigits*).

[3] We use the default tolerance of termination criterion, 0.001, for most of the datasets except *image segmentation* and *shuttle*, which required a less sensitive criterion to converge.

We compare the average loss of the prediction methods aggregated over all of the datasets in Figure 8, showing that on average our method provides lower cost predictions. It is important to note that as an ensemble method, boosting is able to implicitly consider a much richer feature space than our approach. For classification, SVMs are often only comparable when incorporating kernels that can also implicitly consider richer feature spaces. Thus, exceeding the performance of the state-of-the-art boosting method using only quadratic features is a significant demonstration of our method. The comparisons with the structured SVM method, which considers an identical feature space, illustrates the general benefit our approach provides by adversarially approximating the training data rather than convexly approximating the loss function.

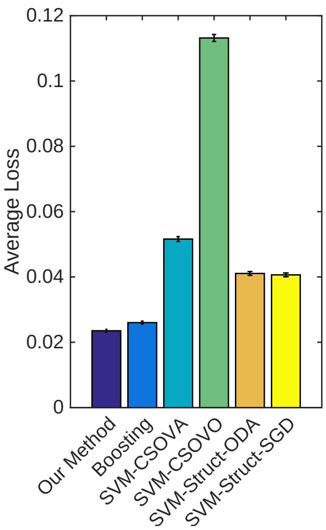

Figure 8: Average loss of predictions across all datasets of Table 1.

5 CONCLUSIONS

In this paper, we have developed an approach for minimizing the exact cost-sensitive loss using an adversarial formulation. In stark contrast with existing methods, which typically minimize a convex approximation of the cost-sensitive loss evaluated on available training data, our approach directly minimizes the actual cost-sensitive loss evaluated on an approximation of the training data. This perspective of placing uncertainty around the training data and resolving it by considering an adversarial evaluator leads to a zero-sum game formulation for inference and convex optimization for estimating model parameters.

We demonstrated the benefits of the approach on a total of 130 prediction tasks. Our approach performs competitively with a state-of-the-art boosting method across many of these tasks and better on average. This is despite the fact that boosting, as an ensemble method, is able to implicitly consider a richer feature space for the classifiers that it ultimately produces. The performance of our approach is much more significantly better than structured multi-class SVM methods and reduction-based SVM methods, which are more directly comparable as they employ the same quadratic feature space.

Our future work will investigate avenues for improving and expanding this adversarial approach to cost-sensitive learning. Foremost, we plan investigate the feasibility of incorporating kernel methods with our approach so that much larger or infinite feature spaces can be tractably incorporated into our cost-sensitive classifier. Additionally, we plan to investigate settings with cost functions that depend on the input attributes in addition to the predicted and actual labels.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. #1227495, *Purposeful Prediction: Co-robot Interaction via Understanding Intent and Goals*.

References

[Abe et al., 2004] Abe, N., Zadrozny, B., and Langford, J. (2004). An iterative method for multi-class cost-sensitive learning. In *KDD*, pages 3–11. ACM.

[Beijbom et al., 2014] Beijbom, O., Saberian, M., Kriegman, D., and Vasconcelos, N. (2014). Guess-averse loss functions for cost-sensitive multiclass boosting. In *Proc. International Conference on Machine Learning*, pages 586–594.

[Bottou et al., 1994] Bottou, L., Cortes, C., Denker, J. S., Drucker, H., Guyon, I., Jackel, L. D., LeCun, Y., Muller, U. A., Sackinger, E., Simard, P., and Vapnik, V. N. (1994). Comparison of classifier methods: a case study in handwritten digit recognition. In *International Conference on Pattern Recognition*, pages 77–82.

[Boyd and Vandenberghe, 2004] Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

[Branson et al., 2013] Branson, S., Beijbom, O., and Belongie, S. (2013). Efficient large-scale structured learning. In *Computer Vision and Pattern Recognition*, pages 1806–1813. IEEE.

[Brefeld et al., 2003] Brefeld, U., Geibel, P., and Wysotzki, F. (2003). Support vector machines with example dependent costs. In *ECML*, pages 23–34. Springer.

[Chan and Stolfo, 1998] Chan, P. K. and Stolfo, S. J. (1998). Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection. In *KDD*, pages 164–168.

[Chang and Lin, 2011] Chang, C.-C. and Lin, C.-J. (2011). Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):1–27.

[Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

[Dalvi et al., 2004] Dalvi, N., Domingos, P., Sanghai, S., Verma, D., et al. (2004). Adversarial classification. In *KDD*, pages 99–108. ACM.

[Davis et al., 2006] Davis, J. V., Ha, J., Rossbach, C. J., Ramadan, H. E., and Witchel, E. (2006). Cost-sensitive decision tree learning for forensic classification. In *ECML*, pages 622–629. Springer.

[Domingos, 1999] Domingos, P. (1999). Metacost: A general method for making classifiers cost-sensitive. In *KDD*, pages 155–164. ACM.

[Elkan, 2001] Elkan, C. (2001). The foundations of cost-sensitive learning. In *IJCAI*, pages 973–978.

[Fan et al., 1999] Fan, W., Stolfo, S. J., Zhang, J., and Chan, P. K. (1999). Adacost: misclassification cost-sensitive boosting. In *ICML*, pages 97–105.

[Freund and Schapire, 1997] Freund, Y. and 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.

[Grünwald and Dawid, 2004] Grünwald, P. D. and Dawid, A. P. (2004). Game theory, maximum entropy, minimum discrepancy, and robust Bayesian decision theory. *Annals of Statistics*, 32:1367–1433.

[Hoffgen et al., 1995] Hoffgen, K.-U., Simon, H.-U., and Vanhorn, K. S. (1995). Robust trainability of single neurons. *Journal of Computer and System Sciences*, 50(1):114–125.

[Knerr et al., 1990] Knerr, S., Personnaz, L., and Dreyfus, G. (1990). Single-layer learning revisited: a stepwise procedure for building and training a neural network. In *Neurocomputing*, pages 41–50. Springer.

[Knoll et al., 1994] Knoll, U., Nakhaeizadeh, G., and Tausend, B. (1994). Cost-sensitive pruning of decision trees. In *ECML*, pages 383–386. Springer.

[Lanckriet et al., 2003] Lanckriet, G. R., Ghaoui, L. E., Bhattacharyya, C., and Jordan, M. I. (2003). A robust minimax approach to classification. *JMLR*, 3:555–582.

[Lee et al., 2004] Lee, Y., Lin, Y., and Wahba, G. (2004). Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association*, 99(465):67–81.

[Lin, 2008] Lin, H.-T. (2008). *From ordinal ranking to binary classification*. PhD thesis, California Institute of Technology.

[Lin, 2010] Lin, H.-T. (2010). A simple cost-sensitive multiclass classification algorithm using one-versus-one comparisons. *National Taiwan University, Tech. Rep.*

[Ling et al., 2004] Ling, C. X., Yang, Q., Wang, J., and Zhang, S. (2004). Decision trees with minimal costs. In *ICML*, pages 544–551. ACM.

[Liu and Ziebart, 2014] Liu, A. and Ziebart, B. D. (2014). Robust classification under sample selection bias. In *Advances in Neural Information Processing Systems*, pages 37–45.

[Lomax and Vadera, 2013] Lomax, S. and Vadera, S. (2013). A survey of cost-sensitive decision tree induction algorithms. *ACM Computing Surveys*, 45(2):16.

[Margineantu, 2002] Margineantu, D. D. (2002). Class probability estimation and cost-sensitive classification decisions. In *ECML*, pages 270–281. Springer.

[Qin et al., 2013] Qin, Z., Wang, A. T., Zhang, C., and Zhang, S. (2013). Cost-sensitive classification with k-nearest neighbors. In *Knowledge Science, Engineering and Management*, pages 112–131. Springer.

[Ratliff et al., 2007] Ratliff, N. D., Bagnell, J. A., and Zinkevich, M. (2007). (approximate) subgradient methods for structured prediction. In *AISTATS*, pages 380–387.

[Savage, 1951] Savage, L. J. (1951). The theory of statistical decision. *Journal of the American Statistical association*, 46(253):55–67.

[Shalev-Shwartz et al., 2011] Shalev-Shwartz, S., Singer, Y., Srebro, N., and Cotter, A. (2011). Pegasos: Primal estimated sub-gradient solver for SVM. *Mathematical programming*, 127(1):3–30.

[Ting, 2000] Ting, K. M. (2000). A comparative study of cost-sensitive boosting algorithms. In *ICML*.

[Topsøe, 1979] Topsøe, F. (1979). Information theoretical optimization techniques. *Kybernetika*, 15(1):8–27.

[Tsochantaridis et al., 2004] Tsochantaridis, I., Hofmann, T., Joachims, T., and Altun, Y. (2004). Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the twenty-first international conference on Machine learning*, page 104. ACM.

[Tsochantaridis et al., 2005] Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y. (2005). Large margin methods for structured and interdependent output variables. In *JMLR*, pages 1453–1484.

[Turney, 1995] Turney, P. D. (1995). Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. *Journal of artificial intelligence research*, pages 369–409.

[von Neumann and Morgenstern, 1947] von Neumann, J. and Morgenstern, O. (1947). *Theory of Games and Economic Behavior*. Princeton University Press.

[Wainwright and Jordan, 2008] Wainwright, M. J. and Jordan, M. I. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305.

[Wald, 1949] Wald, A. (1949). Statistical decision functions. *The Annals of Mathematical Statistics*, 20(2):165–205.

[Wang and Tang, 2012] Wang, R. and Tang, K. (2012). Minimax classifier for uncertain costs. *arXiv preprint arXiv:1205.0406*.

[Wolfowitz, 1950] Wolfowitz, J. (1950). Minimax estimates of the mean of a normal distribution with known variance. *The Annals of Mathematical Statistics*, pages 218–230.

[Zadrozny et al., 2003] Zadrozny, B., Langford, J., and Abe, N. (2003). Cost-sensitive learning by cost-proportionate example weighting. In *ICDM*, pages 435–442.

[Zhou and Liu, 2010] Zhou, Z.-H. and Liu, X.-Y. (2010). On multi-class cost-sensitive learning. *Computational Intelligence*, 26(3):232–257.