# Consistent Robust Adversarial Prediction for General Multiclass Classification

**Rizal Fathony** RFATHO2@UIC.EDU
*Department of Computer Science, University of Illinois at Chicago*

**Kaiser Asif** KASIF2@UIC.EDU
*Department of Computer Science, University of Illinois at Chicago*

**Anqi Liu** ANQILIU@CALTECH.EDU
*Department of Computing and Mathematical Sciences, California Institute of Technology*

**Mohammad Ali Bashiri** MBASHI4@UIC.EDU
*Department of Computer Science, University of Illinois at Chicago*

**Wei Xing** WXING3@UIC.EDU
*Department of Computer Science, University of Illinois at Chicago*

**Sima Behpour** SBEHPO2@UIC.EDU
*Department of Computer Science, University of Illinois at Chicago*

**Xinhua Zhang** ZHANGX@UIC.EDU
*Department of Computer Science, University of Illinois at Chicago*

**Brian D. Ziebart** BZIEBART@UIC.EDU
*Department of Computer Science, University of Illinois at Chicago*

**Editor:**

## Abstract

We propose a *robust adversarial prediction framework* for general multiclass classification. Our method seeks predictive distributions that robustly optimize non-convex and non-continuous multiclass loss metrics against the worst-case conditional label distributions (the adversarial distributions) that (approximately) match the statistics of the training data. Although the optimized loss metrics are non-convex and non-continuous, the dual formulation of the framework is a convex optimization problem that can be recast as a risk minimization model with a prescribed convex surrogate loss we call *the adversarial surrogate loss*. We show that the adversarial surrogate losses fill an existing gap in surrogate loss construction for general multiclass classification problems, by simultaneously aligning better with the original multiclass loss, guaranteeing Fisher consistency, enabling a way to incorporate rich feature spaces via the kernel trick, and providing competitive performance in practice.

**Keywords:** adversarial prediction, multiclass classification, surrogate loss, Fisher consistency, robust distribution.

## 1. Introduction

Multiclass classification is a canonical machine learning task in which a predictor chooses a predicted label from a finite number of possible class labels. For many application domains,

the penalty for making an incorrect prediction is defined by a loss function that depends on the value of the predicted label and the true label. Zero-one loss classification where the predictor suffers a loss of one when making incorrect prediction and zero otherwise and ordinal classification (also known as ordinal regression) where the predictor suffers a loss that increases as the prediction moves farther away from the true label are the examples of the multiclass classification problems.

Empirical risk minimization (ERM) (Vapnik, 1992) is a standard approach for solving general multiclass classification problems by finding the classifier that minimizes a loss metric over the training data. However, since directly minimizing this loss over training data within the ERM framework is generally NP-hard (Steinwart and Christmann, 2008), convex surrogate losses that can be efficiently optimized are employed to approximate the loss. Constructing surrogate losses for binary classification has been well studied, resulting in surrogate losses that enjoy desirable theoretical properties and good performance in practice. Among the popular examples are the logarithmic loss, which is minimized by the logistic regression classifier (McCullagh and Nelder, 1989), and the hinge loss, which is minimized by the support vector machine (SVM) (Boser et al., 1992; Cortes and Vapnik, 1995). Both of these surrogate losses are Fisher consistent (Lin, 2002; Bartlett et al., 2006) for binary classification, meaning they minimize the zero-one loss and yield the Bayes optimal decision when they learn from any true distribution of data using a sufficiently rich feature representation. SVMs provide the additional advantage that when combined with kernel methods, extremely rich feature representations can be efficiently incorporated.

Unfortunately, generalizing the hinge loss to multiclass classification tasks with more than two labels in a theoretically-sound manner is challenging. In the case of multiclass zero-one loss for example, existing extensions of the hinge loss to multiclass convex surrogates (Crammer and Singer, 2002; Weston et al., 1999; Lee et al., 2004) tend to lose their Fisher consistency guarantees (Tewari and Bartlett, 2007; Liu, 2007) or produce low accuracy predictions in practice (Doğan et al., 2016). In the case of multiclass ordinal classification, surrogate losses are usually constructed by transforming the binary hinge loss to take into account the different penalties of the ordinal regression problem using thresholding methods (Shashua and Levin, 2003; Chu and Keerthi, 2005; Lin and Li, 2006; Rennie and Srebro, 2005; Li and Lin, 2007), or sample re-weighting methods (Li and Lin, 2007). Many methods for other general multiclass problems also rely on similar transformations of the binary hinge loss to construct convex surrogates (Binder et al., 2012; Ramaswamy et al., 2018; Lin, 2014). Empirical evaluations have compared the appropriateness of different surrogate losses for general multiclass classification, but these still leave the possibility of undiscovered surrogates that align better with the original multiclass classification loss.

To address these limitations, we propose a *robust adversarial prediction framework* that seeks the most robust (Grünwald and Dawid, 2004; Delage and Ye, 2010) prediction distribution that minimizes the loss metric in the worst-case given statistical summaries of the empirical distributions. We replace the empirical training data for evaluating our predictor with an adversary that is free to choose an evaluating distribution from the set of distributions that (approximately) match the statistical summaries of empirical training data via moment matching constraints of the features. Although the optimized loss metrics are non-convex and non-continuous, we show that the dual formulation of the framework is a

convex empirical risk minimization model with a prescribed convex surrogate loss that we call the *adversarial surrogate loss*.

We develop algorithms to compute the adversarial surrogate losses efficiently: linear time for ordinal classification with the absolute loss metric, quasilinear time for the zero-one loss metric, and linear program-based algorithm for more general loss metrics. We show that the adversarial surrogate losses fill the existing gap in surrogate loss construction for general multiclass classification problems by simultaneously: (1) aligning better with the original multiclass loss metric, since optimizing the surrogate loss is equivalent with optimizing the original loss metric in the primal adversarial prediction formulation; (2) guaranteeing Fisher consistency; (3) enabling computational efficiency in a rich feature representation via the kernel trick; and (4) providing competitive performance in practice.

### 1.1 Contributions of the Paper

Some of the contents in this paper have previously appeared in machine learning conferences: the adversarial prediction formulation for general loss matrices (Asif et al., 2015), the adversarial surrogate loss for the multiclass zero-one loss metric (Fathony et al., 2016), the adversarial surrogate loss for ordinal classification with the absolute loss metric (Fathony et al., 2017), and the Fisher consistency proof in the case of symmetric loss metrics (Fathony et al., 2018). This paper also contains distinct elements to provide a more general view of the adversarial prediction framework for general multiclass classification that have not previously been presented in the conference papers. The following is a summary of the new contributions included in this paper:

1. A general view of adversarial surrogate losses for general multiclass classification problems (Section 3);
2. A new proof technique for deriving the corresponding surrogate loss for a given loss metrics to optimize, based on the extreme points enumeration of the convex polytope (proofs in Section 3);
3. An extension to the ordinal classification problem using the squared loss rather than the absolute loss (Section 3.3);
4. An analysis of the adversarial surrogate loss for the weighted loss metrics (Section 3.4);
5. The loss formulation and prediction scheme of the adversarial surrogate loss for the task of classification with abstention (Section 3.5, Section 4.3);
6. A Fisher consistency analysis for non-symmetric loss metrics under potential-based prediction schemes (Section 5.1);
7. A Fisher consistency analysis for the case where the set of the predictor's options are different from the set of ground truth labels (Section 5.2);
8. A primal optimization algorithm to incorporate rich feature spaces via the kernel trick based on the PEGASOS algorithm (Section 6.2); and
9. Additional experiments for the classification with abstention tasks (Section 7.3).

### 1.2 Paper Organization

This article is organized as follows. The next section formulates the general multiclass classification problem, demonstrates some example problems, and discusses related techniques that solve these problems. Section 3 presents our adversarial prediction framework formulation, and the adversarial surrogate losses constructed from the dual formulation of the framework for several loss metrics including the zero-one loss, absolute loss, squared loss, and abstention-based loss metrics. Section 4 presents two different schemes for making predictions, probabilistic and non-probabilistic schemes. Section 5 establishes the Fisher consistency property of adversarial surrogate losses. Section 6 presents algorithms for optimizing the adversarial surrogate losses as well as the technique to incorporate the kernel trick into the algorithm. Finally, Section 7 discusses experimental evaluations and the empirical advantages of the adversarial surrogate losses compared to the state-of-the-art techniques that can be viewed as risk minimization methods with piece-wise convex surrogates. This includes the generalization of hinge loss and SVM to general multiclass classification problems.

## 2. Preliminaries and Related Works

In multiclass classification problems, the predictor needs to predict a variable by choosing one class from a finite set of possible class labels. The most popular form of multiclass classification uses zero-one loss metric minimization as the objective. This loss metric penalizes all mistakes equally with a loss of one for incorrect predictions and zero loss otherwise. In fact, the term "multiclass classification" itself, is widely used to refer to this specific variant that uses the zero-one loss as the objective. We refer to "general multiclass classification" as the multiclass classification task that can use any loss metric defined based on the predictor's label prediction and the true label in this work.

### 2.1 General Multiclass Classification

In a general multiclass classification problem, the predictor is provided with training examples that are pairs of training data and labels $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ drawn i.i.d. from a distribution $D$ on $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is the feature space and $\mathcal{Y} = [k] \triangleq \{1, \ldots, k\}$ is a finite set of class labels. For a given data point $\mathbf{x}$, the predictor has to provide a class label prediction $\hat{y} \in \mathcal{T} = [l] \triangleq \{1, \ldots, l\}$. Although the set of prediction labels $\mathcal{T}$ is usually the same as the set of ground truth labels $\mathcal{Y}$, we also consider settings in which they differ. A multiclass loss metric $\text{loss}(\hat{y}, y) : \mathcal{T} \times \mathcal{Y} \to [0, \infty)$, denotes the loss incurred by predicting $\hat{y}$ when the true label is $y$. The loss metric, $\text{loss}(\hat{y}, y)$, is also commonly written as a loss matrix $\mathbf{L} \in \mathbb{R}_+^{l \times k}$ (in this case, $\mathbb{R}_+$ refers to $[0, \infty)$), where the value of a matrix cell in $i$-th row and $j$-th column corresponds to the value of $\text{loss}(\hat{y}, y)$ when $\hat{y} = i$ and $y = j$. Some examples of the loss metrics for general multiclass classification problems are:

1. **Zero-one loss metric**. The predictor suffers one loss if its prediction is not the same as the true label, otherwise it suffers zero loss, $\text{loss}^{\text{0-1}}(\hat{y}, y) = I(\hat{y} \neq y)$.

2. **Ordinal classification with absolute loss metric**. The predictor suffers a loss that increases as the prediction moves farther away from the true label. A canonical example for ordinal classification loss metric is the absolute loss, $\text{loss}^{\text{ord}}(\hat{y}, y) = |\hat{y} - y|$.

3. **Ordinal classification with squared loss metric**. The squared loss metric, $\text{loss}^{\text{sq}}(\hat{y}, y) = (\hat{y} - y)^2$, is also popular for evaluating ordinal classification predictions.

4. **Classification with abstention**. In this prediction setting, a standard zero-one loss metric is used. However, the predictor has an additional prediction option to abstain from making a label prediction. Hence, $\mathcal{T} \neq \mathcal{Y}$ in this setting. A constant penalty $\alpha$ is incurred whenever the predictor chooses to use the abstain option.

Example loss matrices for these classification problems are shown in Figure 1.

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix} \quad \text{(a)} \qquad \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 2 & 3 \\ 2 & 1 & 0 & 1 & 2 \\ 3 & 2 & 1 & 0 & 1 \\ 4 & 3 & 2 & 1 & 0 \end{bmatrix} \quad \text{(b)} \qquad \begin{bmatrix} 0 & 1 & 4 & 9 & 16 \\ 1 & 0 & 1 & 4 & 9 \\ 4 & 1 & 0 & 1 & 4 \\ 9 & 4 & 1 & 0 & 1 \\ 16 & 9 & 4 & 1 & 0 \end{bmatrix} \quad \text{(c)}$$

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad \text{(d)}$$

Figure 1: Examples of the loss matrices for general multiclass classification when the number of class labels is 5 and the loss metric is: the zero-one loss (a), ordinal regression with the absolute loss (b), ordinal regression with the squared loss (c), and classification with abstention and $\alpha = \frac{1}{2}$ (d).

### 2.2 Empirical Risk Minimization and Fisher Consistency

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

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

with a regularization penalty $\lambda||\theta||$ often added to avoid overfitting to available training data[1]. Unfortunately, many combinations of classification functions, $\hat{y}_\theta(\mathbf{x})$, and loss metrics, 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).

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

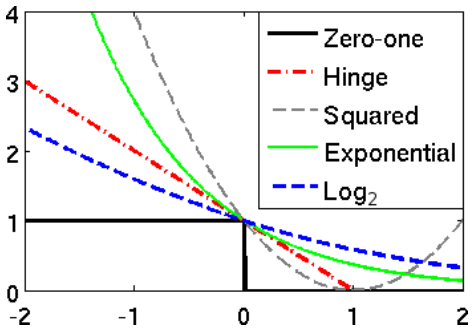


Figure 2: Convex surrogates for the zero-one loss.

To avoid these intractabilities, convex surrogate loss functions (Figure 2) that serve as upper bounds on the desired loss metric are often used to create tractable optimization objectives. 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. The logistic regression classifier (McCullagh and Nelder, 1989) performs a probabilistic prediction by minimizing the logarithmic loss, whereas Adaboost (Freund and Schapire, 1997) incrementally minimizes the exponential loss.

There are many ways to construct convex surrogate loss functions for a given loss metric that we want to optimize. An important property for theoretically guaranteeing optimal prediction is Fisher consistency. It requires a learning method to produce Bayes optimal predictions which minimize the expected loss of this distribution, $\hat{y} \in \text{argmax}_{y'} \mathbb{E}_{Y \sim P}[\text{loss}(y', Y)]$ under ideal learning conditions (trained from the true data distribution $P(Y|\mathbf{X})$ using a fully expressive feature representation). Fisher consistency property guarantees that a learning algorithm (i.e. surrogate loss) reaches the optimal prediction under the original loss metric in the limit. Tewari and Bartlett (2007) presented techniques to characterize the Fisher consistency of surrogate losses for the multiclass zero-one loss metric, which then is extended by Ramaswamy and Agarwal (2012, 2016) to general multiclass loss metrics.

### 2.3 Multiclass Classification Methods

A variety of methods have been proposed to address the general multiclass classification problem. Most of the methods can be viewed as optimizing surrogate losses that come from the extension of binary surrogate loss, e.g., hinge loss (used by SVM), logistic loss (used by logistic regression) and exponential loss (used by AdaBoost), to general multiclass cases. We narrow our focus over this broad range of methods found in the related work to those that can be viewed as empirical risk minimization methods with piece-wise convex surrogates (i.e. generalized hinge loss / generalized SVM), which are more closely related to our approach.

#### 2.3.1 Multiclass Zero-one Classification

The multiclass support vector machine (SVM) seeks class-based potentials $f_y(\mathbf{x})$ for each input vector $\mathbf{x} \in \boldsymbol{\mathcal{X}}$ and class $y \in \mathcal{Y}$ so that the discriminant function, $\hat{y}_{\mathbf{f}}(\mathbf{x}) = \text{argmax}_y f_y(\mathbf{x})$,

minimizes misclassification errors, $\text{loss}_{\mathbf{f}}(\mathbf{x}, y) = I(y \neq \hat{y}_{\mathbf{f}}(\mathbf{x}))$. Many methods have been proposed to generalize SVM to the multiclass setting. Apart from the one-vs-all and one-vs-one decomposed formulations (Deng et al., 2012), there are three main joint formulations:

1. The WW model by Weston et al. (1999), which incorporates the sum of hinge losses for all alternative labels,

$$\text{loss}_{\text{WW}}(\mathbf{x}, y) = \sum_{j \neq y} [1 + (f_j(\mathbf{x}) - f_y(\mathbf{x}))]_+;$$

2. The CS model by Crammer and Singer (2002), which uses the hinge loss of only the largest alternative label,

$$\text{loss}_{\text{CS}}(\mathbf{x}, y) = \max_{j \neq y} [1 + (f_j(\mathbf{x}) - f_y(\mathbf{x}))]_+ ; \text{ and}$$

3. The LLW model by Lee et al. (2004), which employs an absolute hinge loss,

$$\text{loss}_{\text{LLW}}(\mathbf{x}, y) = \sum_{j \neq y} [1 + f_j(\mathbf{x})]_+ ,$$

   and a constraint that $\sum_j f_j(\mathbf{x}) = 0$.

The former two models (CS and WW) both utilize the pairwise class-based potential differences $f_j(\mathbf{x}) - f_y(\mathbf{x})$ and are therefore categorized as relative margin methods. LLW, on the other hand, is an absolute margin method that only relates to $f_j(\mathbf{x})$ (Doğan et al., 2016).

Fisher consistency, or Bayes consistency (Bartlett et al., 2006; Tewari and Bartlett, 2007), guarantees that minimization of a surrogate loss under the true distribution provides the Bayes-optimal classifier, i.e., minimizes the zero-one loss. Among these methods, only the LLW method is Fisher consistent (Lee et al., 2004; Tewari and Bartlett, 2007; Liu, 2007). However, as pointed out by Doğan et al. (2016), LLW's use of an absolute margin in the loss (rather than the relative margin of WW and CS) often causes it to perform poorly for datasets with low dimensional feature spaces. From the opposite direction, the requirements for Fisher consistency have been well-characterized (Tewari and Bartlett, 2007), yet this has not led to a multiclass classifier that is Fisher consistent and performs well in practice.

### 2.3.2 Multiclass Ordinal Classification

Existing techniques for ordinal classification that optimize piece-wise convex surrogates can be categorized into three groups as follows.

1. Threshold methods for ordinal classification.
   Threshold methods treat the ordinal response variable, $\hat{f} \triangleq \mathbf{w} \cdot \mathbf{x}$, as a continuous real-valued variable and introduce $k-1$ thresholds $\eta_1, \eta_2, ..., \eta_{k-1}$ that partition the real line into $k$ segments: $\eta_0 = -\infty < \eta_1 < \eta_2 < ... < \eta_{k-1} < \eta_k = \infty$. Each segment corresponds to a label with $\hat{y}_i$ assigned label $j$ if $\eta_{j-1} < \hat{f} \leq \eta_j$. There are two different approaches for constructing surrogate losses based on the threshold methods to optimize the choice of $\mathbf{w}$ and $\eta_1, \ldots, \eta_{k-1}$ (Shashua and Levin, 2003; Chu and Keerthi, 2005; Rennie and Srebro, 2005). *All thresholds* method (also called SVORIM) penalizes all thresholds involved when a mistake is made. *Immediate thresholds* (also called SVOREX) only penalizes the most immediate thresholds.

2. A reduction framework from ordinal classification to binary classification.
   Li and Lin (2007) proposed a reduction framework to convert ordinal regression problems to binary classification problems by extending training examples. For each training sample $(\mathbf{x}, y)$, the reduction framework creates $k-1$ extended samples $(\mathbf{x}^{(j)}, y^{(j)})$ and assigns weight $w_{y,j}$ to each extended sample. The binary label associated with the extended sample is equivalent to the answer of the question: "is the rank of $\mathbf{x}$ greater than $j$?" The reduction framework allows a choice for how extended samples $\mathbf{x}^{(j)}$ are constructed from original samples $\mathbf{x}$ and how to perform binary classification.

3. Cost-sensitive classification methods for ordinal classification.
   Rather than using thresholding or the reduction framework, ordinal regression can also be cast as a special case of cost-sensitive multiclass classification. Two of the most popular classification-based ordinal regression techniques are extensions of one-versus-one (OVO) and one-versus-all (OVA) cost-sensitive classification (Lin, 2008, 2014). Both algorithms leverage a transformation that converts a cost-sensitive classification problem to a set of weighted binary classification problems. Rather than reducing to binary classification, Tu and Lin (2010) reduce cost-sensitive classification to one-sided regression (OSR), which can be viewed as an extension of the one-versus-all (OVA) technique.

A recent analysis by Pedregosa et al. (2017) shows that many surrogate losses for ordinal classification enjoy Fisher consistency. For example, the *all thresholds* and *immediate thresholds* methods are Fisher consistent provided that the base binary surrogate losses they use are convex with differentiability and a negative derivative at zero.

#### 2.3.3 Multiclass Classification with Abstention

In the classification with abstention setting, a standard zero-one loss is used to evaluate the prediction. However, the predictor has an additional option to abstain from making a label prediction and suffer a constant penalty $\alpha$. In the literature, this type of prediction setting is also called "*classification with reject option*".

Most of the early papers on classification with abstention focused on the binary prediction case. Bartlett and Wegkamp (2008) proposed a consistent surrogate loss based on the SVM's hinge loss for binary classification with abstention where the value of $\alpha$ is restricted to the interval $[0, \frac{1}{2}]$. Grandvalet et al. (2009) extended the approach to the case where the abstention penalty between the positive class $\alpha_+$ and negative class $\alpha_-$ is non-symmetric. A recent study by Cortes et al. (2016) proposed a modification of the boosting algorithm (Freund and Schapire, 1997) that incorporate the abstention setting into the prediction. They also proposed a base weak classifier, *abstention stump*, which is a modification from the popular weak classifier for the standard boosting algorithm (decision stump).

For the multiclass classification setting, a recent paper by Ramaswamy et al. (2018) proposed several algorithms that extend the binary hinge loss to the case of multiclass classification with abstention. They extended the definition of SVM's one-versus-all (OVA) and Crammer-Singer (CS) models to incorporate the abstention penalty. They also proposed a consistent algorithm for multiclass classification with abstention in the case of $\alpha \in [0, \frac{1}{2}]$,

by encoding the prediction classes in binary number representation and formulate a binary encoded prediction (BEP) surrogate.

## 3. Adversarial Prediction Formulation

In a general multiclass classification problem, the predictor needs to make a label prediction $\hat{y} \in \mathcal{T} = \{1, \ldots, l\}$ for a given data point $\mathbf{x}$. To evaluate the performance of the prediction, we compute the multiclass loss metric $\text{loss}(\hat{y}, y)$ by comparing the prediction to the ground truth label $y$. The predictor is also allowed to make a probabilistic prediction by outputting a conditional probability $\hat{P}(\hat{Y}|\mathbf{x})$. In this case, the expected loss $\mathbb{E}_{\hat{Y}|\mathbf{x}\sim\hat{P}}\ \text{loss}(\hat{Y}, y) = \sum_{i=1}^{l} \hat{P}(\hat{Y} = i|\mathbf{x})\ \text{loss}(i, y)$ is measured. Note that in our notation, the upper case $Y$ and $\mathbf{X}$ refer to random variables (of a scalar and vector respectively) while lower case $y$ and $\mathbf{x}$ refer to the observed variables.

Our approach seeks a predictor that robustly minimizes a multiclass loss metric against the worst-case distribution that (approximately) matches the statistics of the training data. In this setting, a predictor makes a probabilistic prediction over the set of all possible labels (denoted as $\hat{P}(\hat{Y}|\mathbf{X})$). Instead of evaluating the predictor with the empirical distribution, the predictor is pitted against an adversary that also makes a probabilistic prediction (denoted as $\check{P}(\check{Y}|\mathbf{X})$). The predictor's objective is to minimize the expected loss metric calculated from the predictor's and adversary's probabilistic predictions, while the adversary seeks to maximize the loss. The adversary is constrained to select a probabilistic prediction that matches the statistical summaries of the empirical training distribution (denoted as $\tilde{P}$) via moment-matching constraints on the features $\phi(\mathbf{x}, y)$.

**Definition 1** *In the* **adversarial prediction framework** *for general multiclass classification, the predictor player first selects a predictive distribution, $\hat{P}(\hat{Y}|\mathbf{X})$, for each input $\mathbf{x}$, from the conditional probability simplex, and then the adversarial player selects an evaluation distribution, $\check{P}(\check{Y}|\mathbf{X})$, for each input $\mathbf{x}$ from the set of distributions consistent with the known statistics:*

$$\min_{\hat{P}(\hat{Y}|\mathbf{X})} \max_{\check{P}(\check{Y}|\mathbf{X})} \mathbb{E}_{\mathbf{X}\sim\tilde{P};\hat{Y}|\mathbf{X}\sim\hat{P};\check{Y}|\mathbf{X}\sim\check{P}} \left[ loss(\hat{Y}, \check{Y}) \right] \tag{1}$$
$$\text{subject to: } \mathbb{E}_{\mathbf{X}\sim\tilde{P};\check{Y}|\mathbf{X}\sim\check{P}}[\phi(\mathbf{X}, \check{Y})] = \mathbb{E}_{\mathbf{X},Y\sim\tilde{P}}\left[\phi(\mathbf{X}, Y)\right].$$

*Here, the statistics $\mathbb{E}_{\mathbf{X},Y\sim\tilde{P}}\left[\phi(\mathbf{X}, Y)\right]$ are a vector of provided feature moments measured from training data.*

For the purpose of establishing efficient learning algorithms, we use the method of Lagrangian multipliers and strong duality for convex-concave saddle point problems (Von Neumann and Morgenstern, 1945; Sion, 1958) to formulate the equivalent dual optimization as stated in Theorem 2.

**Theorem 2** *Determining the value of the* constrained *adversarial prediction minimax game reduces to a minimization over the empirical average of the value of many unconstrained minimax games:*

$$\min_{\theta} \mathbb{E}_{\mathbf{X},Y\sim\tilde{P}} \left[ \max_{\check{P}(\check{Y}|\mathbf{X})} \min_{\hat{P}(\hat{Y}|\mathbf{X})} \mathbb{E}_{\hat{Y}|\mathbf{X}\sim\hat{P};\check{Y}|\mathbf{X}\sim\check{P}} \left[ loss(\hat{Y}, \check{Y}) + \theta^{\intercal} \left( \phi(\mathbf{X}, \check{Y}) - \phi(\mathbf{X}, Y) \right) \right] \right], \tag{2}$$

*where $\theta$ is the Lagrange dual variable for the moment matching constraints.*

**Proof**

$$
\begin{aligned}
&\min_{\hat{P}(\hat{Y}|\mathbf{X})} \max_{\check{P}(\check{Y}|\mathbf{X})} \mathbb{E}_{\mathbf{X}\sim\tilde{P};\hat{Y}|\mathbf{X}\sim\hat{P};\check{Y}|\mathbf{X}\sim\check{P}} \Big[\text{loss}(\hat{Y},\check{Y})\Big] \\
&\qquad \text{subject to: } \mathbb{E}_{\mathbf{X}\sim\tilde{P};\check{Y}|\mathbf{X}\sim\check{P}} \left[\phi(\mathbf{X},\check{Y})\right] = \mathbb{E}_{\mathbf{X},Y\sim\tilde{P}} \left[\phi(\mathbf{X},Y)\right] \\
\overset{(a)}{=} &\max_{\check{P}(\check{Y}|\mathbf{X})} \min_{\hat{P}(\hat{Y}|\mathbf{X})} \mathbb{E}_{\mathbf{X}\sim\tilde{P};\hat{Y}|\mathbf{X}\sim\hat{P};\check{Y}|\mathbf{X}\sim\check{P}} \Big[\text{loss}(\hat{Y},\check{Y})\Big] \\
&\qquad \text{subject to: } \mathbb{E}_{\mathbf{X}\sim\tilde{P};\check{Y}|\mathbf{X}\sim\check{P}} \left[\phi(\mathbf{X},\check{Y})\right] = \mathbb{E}_{\mathbf{X},Y\sim\tilde{P}} \left[\phi(\mathbf{X},Y)\right] \\
\overset{(b)}{=} &\max_{\check{P}(\check{Y}|\mathbf{X})} \min_{\theta} \min_{\hat{P}(\hat{Y}|\mathbf{X})} \mathbb{E}_{\mathbf{X},Y\sim\tilde{P};\hat{Y}|\mathbf{X}\sim\hat{P};\check{Y}|\mathbf{X}\sim\check{P}} \Big[\text{loss}(\hat{Y},\check{Y}) + \theta^\intercal \left(\phi(\mathbf{X},\check{Y}) - \phi(\mathbf{X},Y)\right)\Big] \\
\overset{(c)}{=} &\min_{\theta} \max_{\check{P}(\check{Y}|\mathbf{X})} \min_{\hat{P}(\hat{Y}|\mathbf{X})} \mathbb{E}_{\mathbf{X},Y\sim\tilde{P};\hat{Y}|\mathbf{X}\sim\hat{P};\check{Y}|\mathbf{X}\sim\check{P}} \Big[\text{loss}(\hat{Y},\check{Y}) + \theta^\intercal \left(\phi(\mathbf{X},\check{Y}) - \phi(\mathbf{X},Y)\right)\Big] \\
\overset{(d)}{=} &\min_{\theta} \mathbb{E}_{\mathbf{X},Y\sim\tilde{P}} \left[\max_{\check{P}(\check{Y}|\mathbf{X})} \min_{\hat{P}(\hat{Y}|\mathbf{X})} \mathbb{E}_{\hat{Y}|\mathbf{X}\sim\hat{P};\check{Y}|\mathbf{X}\sim\check{P}} \Big[\text{loss}(\hat{Y},\check{Y}) + \theta^\intercal \left(\phi(\mathbf{X},\check{Y}) - \phi(\mathbf{X},Y)\right)\Big]\right].
\end{aligned}
$$

The transformation steps above are described as follows:

(a) We flip the min and max order using minimax duality (Von Neumann and Morgenstern, 1945). The domains of $\hat{P}(\hat{Y}|\mathbf{X})$ and $\check{P}(\check{Y}|\mathbf{X})$ are both compact convex sets and the objective function is bilinear, therefore, strong duality holds.

(b) We introduce the Lagrange dual variable $\theta$ to directly incorporate the equality constraints into the objective function.

(c) The domain of $\check{P}(\check{Y}|\mathbf{X})$ is a compact convex subset of $\mathbb{R}^n$, while the domain of $\theta$ is $\mathbb{R}^m$. The objective is concave on $\check{P}(\check{Y}|\mathbf{X})$ for all $\theta$ (a non-negative linear combination of minimums of affine functions is concave), while it is convex on $\theta$ for all $\check{P}(\check{Y}|\mathbf{X})$. Based on Sion's minimax theorem (Sion, 1958), strong duality holds, and thus we can flip the optimization order of $\check{P}(\check{Y}|\mathbf{X})$ and $\theta$.

(d) Since the expression is additive in terms of $\check{P}(\check{Y}|\mathbf{X})$ and $\hat{P}(\hat{Y}|\mathbf{X})$, we can push the expectation over the empirical distribution $\mathbf{X}, Y \sim \tilde{P}$ outside and independently optimize each $\check{P}(\check{Y}|\mathbf{x})$ and $\hat{P}(\hat{Y}|\mathbf{x})$.

■

The dual problem (Eq. (2)) possesses the important property of being a **convex** optimization problem in $\theta$. The objective of Eq. (2) consists of the function $\text{loss}(\hat{Y},\check{Y}) + \theta^\intercal \left(\phi(\mathbf{X},\check{Y}) - \phi(\mathbf{X},Y)\right)$ which is an affine function with respect to $\theta$, followed by operations that preserve convexity (Boyd and Vandenberghe, 2004): (1) the non-negative weighted sum (the expectations in the objective), (2) the minimization in the predictor $\hat{P}(\hat{Y}|X)$ over a non-empty convex set out of a function that is jointly convex in $\theta$ and $\hat{P}(\hat{Y}|X)$, and (3) the point-wise maximum in the adversary distribution $\check{P}(\check{Y}|X)$ over an infinite set of convex functions. Therefore, the overall objective is convex with respect to $\theta$. This property

is important since we can use gradient-based optimization in our learning algorithm and guarantee convergence to the global optimum of the objective despite the fact that the original loss metrics we want to optimize in the primal formulation of the adversarial prediction (Eq. (1)) are non-convex and non-continuous.

Despite the different motivations between our adversarial prediction framework and the empirical risk minimization framework, the dual optimization formulation (Eq. (2)) resembles a risk minimization problem with the surrogate loss defined as:

$$AL(\mathbf{x}, y, \theta) = \max_{\check{P}(\check{Y}|\mathbf{x})} \min_{\hat{P}(\hat{Y}|\mathbf{x})} \mathbb{E}_{\hat{Y}|\mathbf{x}\sim\hat{P};\check{Y}|\mathbf{x}\sim\check{P}} \left[\text{loss}(\hat{Y}, \check{Y}) + \theta^\intercal \left(\phi(\mathbf{x}, \check{Y}) - \phi(\mathbf{x}, y)\right)\right]. \quad (3)$$

We call this surrogate loss the "adversarial surrogate loss" or in short "AL". In the next subsections, we will analyze more about this surrogate loss for different instances of general multiclass classification problems.

Let us first simplify the notation used in our surrogate loss. We construct a vector $\mathbf{p}$ to compactly represent the predictor's conditional probability $\hat{P}(\hat{Y}|\mathbf{x})$, where the value of its $i$-th index is $p_i = \hat{P}(\hat{Y} = i|\mathbf{x})$. Similarly, we construct a vector $\mathbf{q}$ for the adversary's conditional probability, i.e., $q_i = \check{P}(\check{Y} = i|\mathbf{x})$. We also define a potential vector $\mathbf{f}$ whose $i$-th index stores the potential for the $i$-th class, i.e., $f_i = \theta^\intercal\phi(\mathbf{x}, i)$. Finally, we use a matrix $\mathbf{L}$ to represent the loss function introduced at the beginning of this section. Using these notations we can rewrite our adversarial surrogate loss as:

$$AL(\mathbf{f}, y) = \max_{\mathbf{q}\in\Delta} \min_{\mathbf{p}\in\Delta} \mathbf{p}^\intercal\mathbf{L}\mathbf{q} + \mathbf{f}^\intercal\mathbf{q} - f_y,$$

where $\Delta$ denotes the conditional probability simplex. The maximin formulation above can be converted to a linear program as follows:

$$\begin{aligned} AL(\mathbf{f}, y) = \max_{\mathbf{q}, v}\ & v + \mathbf{f}^\intercal\mathbf{q} - f_y \\ \text{s.t.: } & \mathbf{L}_{(i,:)}\mathbf{q} \geq v \quad \forall i \in [k] \\ & q_i \geq 0 \qquad\quad \forall i \in [k] \\ & \mathbf{q}^\intercal\mathbf{1} = 1, \end{aligned} \quad (4)$$

where $v$ is a slack variable for converting the inner minimization into sets of linear inequality constraints, and $\mathbf{L}_{(i,:)}$ denote the $i$-th row of matrix $\mathbf{L}$. We will analyze the solution of this linear program for several different types of loss metrics to construct a simpler closed-form formulation of the surrogate loss.

### 3.1 Multiclass Zero-One Classification

The multiclass zero-one loss metric is one of the most popular metrics used in multiclass classification. The loss metric penalizes an incorrect prediction with a loss of one and zero otherwise, i.e., $\text{loss}(\hat{y}, y) = I(\hat{y} \neq y)$. An example of zero-one loss matrix for classification with five classes can be seen in Figure 1a.

We focus on analyzing the solution of the maximization in Eq. (4) for the case where $\mathbf{L}$ is the zero-one loss matrix. Since the objective in Eq. (4) is linear and the constraints form a convex polytope $\mathbb{C}$ over the space of $\begin{bmatrix}\mathbf{q}\\ v\end{bmatrix}$, there is always an optimal solution that is an

extreme point of the domain (Theorem 32.2 of Rockafellar, 1970). The only catch is that $\mathbb{C}$ is not bounded, but this can be easily addressed by adding a nominal constraint $v \geq -1$ (see Proposition 4). Our strategy is to first characterize the extreme points of $\mathbb{C}$ that may possibly solve Eq. (4), and then the evaluation of adversarial loss ($AL$) becomes equivalent to finding an extreme point that maximizes the objective in Eq. (4).

The polytope $\mathbb{C}$ can be defined in its canonical form by using the half-space representation of a polytope as follows:

$$\mathbb{C} = \left\{ \begin{bmatrix} \mathbf{q} \\ v \end{bmatrix} \middle| \mathbf{A} \begin{bmatrix} \mathbf{q} \\ v \end{bmatrix} \geq \mathbf{b}, \quad \text{where} \quad \mathbf{A} = \begin{bmatrix} \mathbf{L} & -\mathbf{1} \\ \mathbf{I} & \mathbf{0} \\ \mathbf{1}^\intercal & 0 \\ -\mathbf{1}^\intercal & 0 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ 1 \\ -1 \end{bmatrix} \right\}. \tag{5}$$

Here $\mathbf{L}$ is a $k$-by-$k$ loss matrix, $\mathbf{I}$ is a $k$-by-$k$ identity matrix, $\mathbf{1}$ and $\mathbf{0}$ are vectors with length $k$ that contain all 1 and or all 0 respectively. $\mathbf{A}$ has $2k+2$ rows and $k+1$ columns. Below is an example of this half-space representation for a four-class classification with zero-one loss metric:

$$\begin{array}{l} \\ \text{1st block} \\ \\ \\ \\ \hdashline \\ \\ \\ \text{2nd block} \\ \\ \hdashline \\ \text{3rd block} \\ \end{array} \begin{bmatrix} 0 & 1 & 1 & 1 & -1 \\ 1 & 0 & 1 & 1 & -1 \\ 1 & 1 & 0 & 1 & -1 \\ 1 & 1 & 1 & 0 & -1 \\ \hdashline 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \hdashline 1 & 1 & 1 & 1 & 0 \\ -1 & -1 & -1 & -1 & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ v \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ -1 \end{bmatrix}.$$

For simplicity, we divide $\mathbf{A}$ into 3 blocks of rows. The first block contains $k$ rows defining the constraints that relate the loss matrix with the slack variable $v$, the second block also contains $k$ rows for non-negativity constraints, and the third block is for the sum-to-one constraints.

To characterize the extreme points of $\mathbb{C}$ that solve Eq. (4), we utilize the algebraic characterization of extreme points in a bounded polytope given by Theorem 3.17 from Andréasson et al. (2005). For convenience, we quote it here.

**Proposition 3 (Theorem 3.17 from Andréasson et al. (2005))** *Let $\mathbb{P} \triangleq \{\mathbf{c} \in \mathbb{R}^n \mid \mathbf{A}\mathbf{c} \geq \mathbf{b}\}$ be a bounded polytope, where $\mathbf{A} \in \mathbb{R}^{m \times n}$ has $rank(\mathbf{A}) = n$ and $\mathbf{b} \in \mathbb{R}^m$. For any $\bar{\mathbf{c}} \in \mathbb{P}$, let $\mathcal{I}(\bar{\mathbf{c}})$ be the set of row index $i$ such that $\mathbf{A}_{(i,:)}\bar{\mathbf{c}} = b_i$. Let $\mathbf{A}_{\bar{\mathbf{c}}}$ and $\mathbf{b}_{\bar{\mathbf{c}}}$ be the submatrix and subvector of $\mathbf{A}$ and $\mathbf{b}$ that extract the rows in $\mathcal{I}(\bar{\mathbf{c}})$, respectively. Then $\mathbf{A}_{\bar{\mathbf{c}}}\mathbf{c} = \mathbf{b}_{\bar{\mathbf{c}}}$ is called the* equality subsystem *for $\bar{\mathbf{c}}$, and $\bar{\mathbf{c}} \in \mathbb{P}$ is an extreme point if and only if $rank(\mathbf{A}_{\bar{\mathbf{c}}}) = n$.*

Since $\mathbb{C}$ is not bounded ($v$ can diverge to $-\infty$), we now further characterize a subset of $\mathbb{C}$ that must include an optimal solution to Eq. (4).

**Proposition 4** *Let ext* $\mathbb{C} = \{\mathbf{c} \in \mathbb{C} | \operatorname{rank}(\mathbf{A_c}) = k+1\}$. *Then ext* $\mathbb{C}$ *must contain an optimal solution to Eq.* (4).

**Proof** Let us add a nominal constraint of $v \geq -1$ to the definition of $\mathbb{C}$, and denote the new polytope as $\bar{\mathbb{C}} := \left\{\mathbf{c} : \mathbf{G}\mathbf{c} \geq \begin{bmatrix}\mathbf{b} \\ -1\end{bmatrix}\right\}$, where $\mathbf{G} = \begin{bmatrix}\mathbf{A} \\ \mathbf{0}^\intercal\ 1\end{bmatrix}$. It does not change the solution to Eq. (4) because $v$ appears in the objective only as $v$, and $\mathbf{L}_{(i,:)}\mathbf{q} \geq 0$. However, this additional constraint makes $\bar{\mathbb{C}}$ compact, allowing us to apply Theorem 3.17 of (Andréasson et al., 2005) and conclude that any $\mathbf{c} = \begin{bmatrix}\mathbf{q} \\ v\end{bmatrix}$ is an extreme point of $\bar{\mathbb{C}}$ if and only if $\operatorname{rank}(\mathbf{G_c}) = k+1$. But all optimal solutions must have $v \geq 0$, hence the last row of $\mathbf{G}$ cannot be in $\mathbf{G_c}$. So it suffices to consider $\mathbf{c}$ with $\mathbf{G_c} = \mathbf{A_c}$, whence $\operatorname{rank}(\mathbf{A_c}) = k+1$. ■

Obviously $\mathbf{A_c}$ must include the third block of $\mathbf{A}$ for all $\mathbf{c} \in \mathbb{C}$ in Eq. (5). The rank condition also enforces that at least one row from the first block is selected.

For convenience, we will refer to ext $\mathbb{C}$ as the extreme point of $\mathbb{C}$.[2] By analyzing ext $\mathbb{C}$ in the case of multiclass zero-one classification, we simplify the adversarial surrogate loss (Eq. (4)) as stated in the following Theorem 5.

**Theorem 5** *The model parameter* $\theta$ *for multiclass zero-one adversarial classification is equivalently obtained from empirical risk minimization under the adversarial zero-one loss function:*

$$AL^{0\text{-}1}(\mathbf{f}, y) = \max_{S \subseteq [k],\, S \neq \emptyset} \frac{\sum_{i \in S} f_i + |S| - 1}{|S|} - f_y, \tag{6}$$

*where* $S$ *is any non-empty subset of the* $k$ *classes.*

**Proof** The $\text{AL}^{0\text{-}1}$ above corresponds to the set of "extreme points"[3]

$$D = \left\{ \begin{bmatrix}\mathbf{q} \\ v\end{bmatrix} = \frac{1}{|S|} \begin{bmatrix}\sum_{i \in S} \mathbf{e}_i \\ |S| - 1\end{bmatrix} \;\middle|\; \emptyset \neq S \subseteq [k] \right\},$$

where $\mathbf{e}_i \in \mathbb{R}^k$ is the $i$-th canonical vector with a single 1 at the $i$-th coordinate and 0 elsewhere. That means $\mathbf{q}$ first picks a nonempty support $S \subseteq [k]$, then places uniform probability of $\frac{1}{|S|}$ on these coordinates, and finally sets $v$ to $\frac{|S|-1}{|S|}$.

By Proposition 4, it now suffices to prove that $D \subseteq \mathbb{C}$ and $D \supseteq \operatorname{ext} \mathbb{C} = \{\mathbf{c} \in \mathbb{C} : \operatorname{rank}(\mathbf{A_c}) = k+1\}$, i.e., any $\mathbf{c} \in \mathbb{C}$ whose equality system satisfies $\operatorname{rank}(\mathbf{A_c}) = k+1$ must be in $D$. $D \subseteq \mathbb{C}$ is trivial, so we focus on $D \supseteq \operatorname{ext} \mathbb{C}$.

Given $\mathbf{c} \in \operatorname{ext} \mathbb{C}$, suppose the set of rows that $\mathbf{A_c}$ selected from the first and second block of $\mathbf{A}$ are $R$ and $T$, respectively. Both $R$ and $T$ are subsets of $[k]$, indexed against $\mathbf{A}$.

2. Indeed, it is the bona fide extreme point set of $\mathbb{C}$ under the standard definition which does not require compactness (Section 18, Rockafellar, 1970). But the guarantee of attaining optimality at an extreme point does require boundedness.
3. We add a quotation mark here because our proof will only show, as it suffices to show, that $D$ contains all the extreme points of $\mathbb{C}$ and $D \subseteq \mathbb{C}$. We do not need to show that $D$ is exactly the extreme point set of $\mathbb{C}$, although that fact is not hard to show either.

We first observe that $R$ and $T$ must be disjoint because if $i \in R \cap T$, then $q_i = 0$ and $v = \mathbf{L}_{(i,:)}\mathbf{q} = \sum_{j \neq i} q_j = 1 - q_i = 1$. But then for all $j$, $\mathbf{L}_{(j,:)}\mathbf{q} \geq v$ implies $1 \leq \sum_{l \neq j} q_l = 1 - q_j$. This is impossible as it means $\mathbf{q} = \mathbf{0}$.

Now that $R$ and $T$ are disjoint, $\text{rank}(\mathbf{A_c}) = k + 1$ implies that $R = [k] \backslash T$. Since $q_i = 0$ for all $i \in T$, solving $|R|$ linear equalities with respect to $|R|$ unknowns yields $q_j = 1/|R|$ for all $j \in R$. Such a tuple of $\mathbf{q}$ and $v$ is clearly in $D$. Obviously $R$ cannot be empty because then $T = [k]$ and $\mathbf{q} = \mathbf{0}$. ■

We denote the potential differences $\psi_{i,y} = f_i - f_y$, then Eq (6), can be equivalently written as:

$$\text{AL}^{\text{0-1}}(\mathbf{f}, y) = \max_{S \subseteq [k],\, S \neq \emptyset} \frac{\sum_{i \in S} \psi_{i,y} + |S| - 1}{|S|}.$$

Thus, $\text{AL}^{\text{0-1}}$ is the maximum value over $2^k - 1$ linear hyperplanes. For binary prediction tasks, there are three linear hyperplanes: $\psi_{1,y}, \psi_{2,y}$ and $\frac{\psi_{1,y}+\psi_{2,y}+1}{2}$. Figure 3 shows the loss function in potential difference space $\psi$ when the true label is $y = 1$. Note that $\text{AL}^{\text{0-1}}$ combines two hinge functions at $\psi_{2,y} = -1$ and $\psi_{2,y} = 1$, rather than SVM's single hinge at $\psi_{2,y} = -1$. This difference from the hinge loss corresponds to the loss that is realized by randomizing label predictions of $\hat{P}(\hat{Y}|\mathbf{x})$ in Eq. (3).

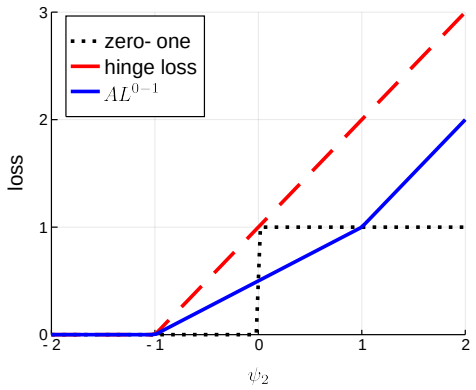


Figure 3: $\text{AL}^{\text{0-1}}$ evaluated over the space of potential differences ($\psi_{i,y} = f_i - f_y$; and $\psi_{i,i} = 0$) for binary prediction tasks when the true label is $y = 1$.

For three classes, the loss function has seven facets as shown in Figure 4a. Figures 4a, 4b, and 4c show the similarities and differences between $\text{AL}^{\text{0-1}}$ and the multiclass SVM surrogate losses based on class potential differences. Note that $\text{AL}^{\text{0-1}}$ is a relative margin loss function that utilizes the pairwise potential *difference* $\psi_{i,y}$. This avoids the surrogate loss construction pitfall pointed out by Doğan et al. (2016) that states that surrogate losses based on the absolute margin (rather than relative margin) may suffer from low performance for datasets with low dimenional feature spaces.

Even though $\text{AL}^{\text{0-1}}$ is the maximization over $2^k - 1$ possible values, it can be efficiently computed as follows. First we need to sort the potential for all labels $\{f_i : i \in [k]\}$ in non-increasing order. The set $S^*$ that maximize $\text{AL}^{\text{0-1}}$ must include the first $j$ labels in the

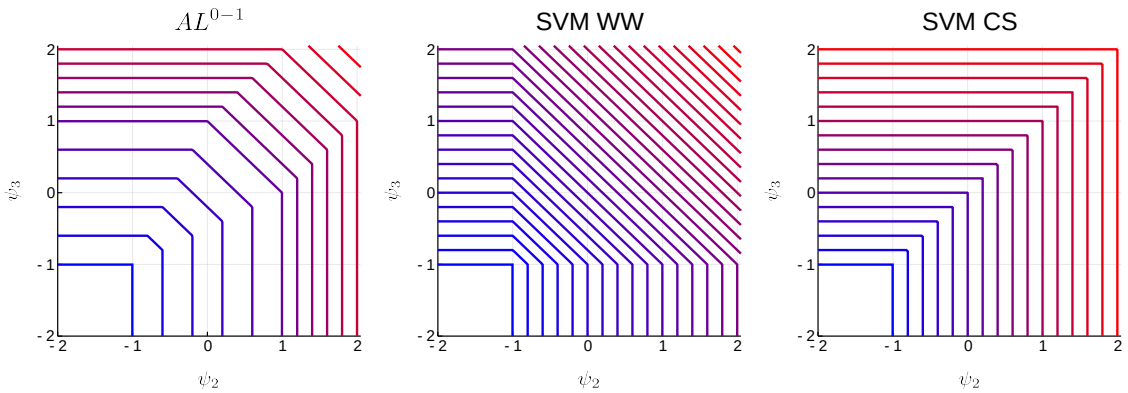


Figure 4: Loss function contour plots over the space of potential differences for the prediction task with three classes when the true label is $y = 1$ under AL$^{0\text{-}1}$ (a), the WW loss (b), and the CS loss (c). (Note that $\psi_i$ in the plots refers to $\psi_{i,y} = f_i - f_y$; and $\psi_{i,i} = 0$.)

sorted order, for some value of $j$. Therefore, to compute AL$^{0\text{-}1}$, we can incrementally add the label in the sorted order to the set $S^*$ until adding an additinal label would decrease the value of the loss.[4] This results in an algorithm with a runtime complexity of $\mathcal{O}(k \log k)$, which is much faster than enumerating all possible values in the maximization.

### 3.2 Ordinal Classification with Absolute Loss

In multiclass ordinal classification (also known as ordinal regression), the discrete class labels being predicted have an inherent order (e.g., *poor*, *fair*, *good*, *very good*, and *excellent* labels). The absolute error, $\text{loss}(\hat{y}, y) = |\hat{y} - y|$ between label prediction ($\hat{y} \in \mathcal{Y}$) and actual label ($y \in \mathcal{Y}$) is a canonical ordinal regression loss metric. The adversarial surrogate loss for ordinal classification using the absolute loss metric is defined in Eq. (4), where $\mathbf{L}$ is the absolute loss matrix (e.g., Figure 1b for a five class ordinal classification). The constraints in Eq. (4) form a convex polytope $\mathbb{C}$. Below is an example of the half-space representation of $\mathbb{C}$ for a four-class ordinal classification problem.

$$
\begin{array}{l}
\\
\text{1st block} \\
\\
\\
\hdashline
\\
\text{2nd block} \\
\\
\\
\hdashline
\\
\text{3rd block} \\
\\
\end{array}
\left[\begin{array}{ccccc}
0 & 1 & 2 & 3 & -1 \\
1 & 0 & 1 & 2 & -1 \\
2 & 1 & 0 & 1 & -1 \\
3 & 2 & 1 & 0 & -1 \\
\hdashline
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
\hdashline
1 & 1 & 1 & 1 & 0 \\
-1 & -1 & -1 & -1 & 0
\end{array}\right]
\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ v \end{bmatrix}
\geq
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ -1 \end{bmatrix}.
$$

4. We refer the reader to the Appendix C of (Fathony et al., 2016) for the optimality proof of this algorithm.

By analyzing the extreme points of $\mathbb{C}$, we define the adversarial surrogate loss for ordinal classification with absolute loss $\text{AL}^{\text{ord}}$ as stated in Theorem 6.

**Theorem 6** *An adversarial ordinal classification predictor with absolute loss is obtained by choosing parameters $\theta$ that minimize the empirical risk of the surrogate loss function:*

$$AL^{ord}(\mathbf{f}, y) = \max_{i,j \in [k]} \frac{f_i + f_j + j - i}{2} - f_y.$$

**Proof** The $\text{AL}^{\text{ord}}$ above corresponds to the set of "extreme points"

$$D = \left\{ \begin{bmatrix} \mathbf{q} \\ v \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \mathbf{e}_i + \mathbf{e}_j \\ j - i \end{bmatrix} \;\middle|\; i, j \in [k] \right\}.$$

This means $\mathbf{q}$ can only have one or two non-zero elements (note that $i$ and $j$ can be equal) with uniform probability of $\frac{1}{2}$ and the value of $v$ is $\frac{j-i}{2}$.

Similar to the proof of Theorem 5, we next prove that $D \supseteq \text{ext}\ \mathbb{C} = \{\mathbf{c} \in \mathbb{C} : \text{rank}(\mathbf{A_c}) = k+1\}$. Given $\mathbf{c} \in \text{ext}\ \mathbb{C}$, suppose the set of rows that $\mathbf{A_c}$ selected from the first and second block of $\mathbf{A}$ are $S$ and $T$, respectively. Both $S$ and $T$ are subsets of $[k]$, indexed against $\mathbf{A}$. Denote $s_{\max} = \max(S)$ and $s_{\min} = \min(S)$. We consider two cases:

1. $S \cap T = \emptyset$: the indices selected from the first and second blocks are disjoint.
   It is easy to check that $\mathbf{c}$ must be $\begin{bmatrix} \mathbf{q} \\ v \end{bmatrix} := \frac{1}{2} \begin{bmatrix} \mathbf{e}_{s_{\max}} + \mathbf{e}_{s_{\min}} \\ s_{\max} - s_{\min} \end{bmatrix}$. Obviously it satisfies (being equal) the rows in $\mathbf{A_c}$ extracted from the first and third blocks of $\mathbf{A}$, because $|l - s_{\max}| + |l - s_{\min}| = s_{\max} - s_{\min}$ for all $l \in S$. Since $S \cap T = \emptyset$, $\mathbf{c}$ must also satisfy those rows from the second block. Finally notice that only one vector in $\mathbb{R}^{k+1}$ can meet all the equalities encoded by $\mathbf{A_c}$ because $\text{rank}(\mathbf{A_c}) = k+1$. Obviously $\mathbf{c} \in D$.

2. $S \cap T \neq \emptyset$: the indices from the first block overlap with those from the second block. Including in $\mathbf{A_c}$ the $i$-th row of the second block means setting $q_i$ to 0. Denote the set of remaining indices as $R = [k] \backslash T$, and let $r_{\max} = \max(R)$ and $r_{\min} = \min(R)$. Now consider two sub-cases:

   a) $r_{\min} \leq s_{\min}$ and $r_{\max} \geq s_{\max}$.

   One may check that $\mathbf{c}$ must be $\begin{bmatrix} \mathbf{q} \\ v \end{bmatrix} := \frac{1}{2} \begin{bmatrix} \mathbf{e}_{r_{\max}} + \mathbf{e}_{r_{\min}} \\ r_{\max} - r_{\min} \end{bmatrix}$. Obviously it satisfies (being equal) the rows in $\mathbf{A_c}$ extracted from the first and third blocks of $\mathbf{A}$, because for all $l \in S$, $l \geq s_{\min} \geq r_{\min}$ and $l \leq s_{\max} \leq r_{\max}$, implying $|l - r_{\max}| + |l - r_{\min}| = r_{\max} - r_{\min}$. Since by definition $r_{\max}$ and $r_{\min}$ are not among the rows selected from the second block, the equalities from the second block must also be satisfied. As in case 1, only one vector in $\mathbb{R}^{k+1}$ can meet all the equalities encoded by $\mathbf{A_c}$ because $\text{rank}(\mathbf{A_c}) = k+1$. Obviously $\mathbf{c} \in D$.

   b) $r_{\min} > s_{\min}$ or $r_{\max} < s_{\max}$.
   We first show $r_{\min} > s_{\min}$ is impossible. By definition of $R$, $q_l = 0$ for all $l < r_{\min}$. For all $l \geq r_{\min}$ ($> s_{\min}$), it follows that $\mathbf{L}_{(s_{\min}, l)} = l - s_{\min} > l - r_{\min} = \mathbf{L}_{(r_{\min}, l)}$.

Noting that at least one $q_l$ must be positive for $l \geq r_{\min}$ (because of the sum-to-one constraint), we conclude that $\mathbf{L}_{(s_{\min},:)}\mathbf{q} > \mathbf{L}_{(r_{\min},:)}\mathbf{q}$. But this contradicts with $\mathbf{L}_{(s_{\min},:)}\mathbf{q} = v \leq \mathbf{L}_{(r_{\min},:)}\mathbf{q}$, where the equality is because $s_{\min} \in S$.
Similarly, $r_{\max} < s_{\max}$ is also impossible.

Therefore, in all possible cases, we have shown that any $\mathbf{c}$ in ext $\mathbb{C}$ must be in $D$. Further noticing the obvious fact that $D \subseteq \mathbb{C}$, we conclude our proof. ■

We note that the $\text{AL}^{\text{ord}}$ surrogate is the maximization over pairs of different potential functions associated with each class (including pairs of identical class labels) added to the distance between the pair. To compute the loss more efficiently, we make use of the fact that maximization over each element of the pair can be independently realized:

$$\max_{i,j\in[k]} \frac{f_i + f_j + j - i}{2} - f_y = \tfrac{1}{2}\max_i (f_i - i) + \tfrac{1}{2}\max_j (f_j + j) - f_y. \tag{7}$$

We derive two different versions of $\text{AL}^{\text{ord}}$ based on different feature representations used for constraining the adversary's probability distribution.

### 3.2.1 Feature Representations

We consider two feature representations corresponding to different training data summaries:

$$\phi_{th}(\mathbf{x}, y) = \begin{pmatrix} y\mathbf{x} \\ I(y \leq 1) \\ I(y \leq 2) \\ \vdots \\ I(y \leq k-1) \end{pmatrix}; \text{ and } \quad \phi_{mc}(\mathbf{x}, y) = \begin{pmatrix} I(y=1)\mathbf{x} \\ I(y=2)\mathbf{x} \\ I(y=3)\mathbf{x} \\ \vdots \\ I(y=k)\mathbf{x} \end{pmatrix}.$$

The first, which we call the **thresholded regression representation**, has size $m+k-1$, where $m$ is the dimension of our input space. It induces a single shared vector of feature weights and a set of thresholds. If we denote the weight vector associated with the $y\mathbf{x}$ term as $\mathbf{w}$ and the terms associated with the cumulative sum of class indicator functions as $\eta_1$, $\eta_2$, …, $\eta_{k-1}$, then thresholds for switching between class $i$ and $i+1$ (ignoring other classes) occur when $\mathbf{w} \cdot \mathbf{x} = \eta_j$.

The second feature representation, $\phi_{mc}$, which we call the **multiclass representation**, has size $mk$ and can be equivalently interpreted as inducing a set of class-specific feature weights, $f_i = \mathbf{w}_i \cdot \mathbf{x}$. This feature representation is useful when ordered labels cannot be thresholded according to any single direction in the input space, as shown in the example dataset of Figure 5.

### 3.2.2 Thresholded regression surrogate loss

In the thresholded regression feature representation, the parameter contains a single shared vector of feature weights $\mathbf{w}$ and $k-1$ terms $\eta_k$ associated with thresholds. Following Eq. (7), the adversarial ordinal regression surrogate loss for this feature representation can be

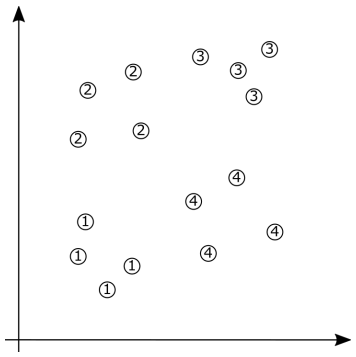

Figure 5: Example where multiple weight vectors are useful.

written as:

$$\mathrm{AL}^{\text{ord-th}}(\mathbf{x}, y) = \max_i \frac{i(\mathbf{w}\cdot\mathbf{x} - 1) + \sum_{l\geq i}\eta_l}{2} + \max_j \frac{j(\mathbf{w}\cdot\mathbf{x} + 1) + \sum_{l\geq j}\eta_l}{2} - y\mathbf{w}\cdot\mathbf{x} - \sum_{l\geq y}\eta_l.$$

This loss has a straight-forward interpretation in terms of the thresholded regression perspective, as shown in Figure 6: it is based on averaging the thresholded label predictions for potentials $\mathbf{w}\cdot\mathbf{x} - 1$ and $\mathbf{w}\cdot\mathbf{x} + 1$. This penalization of the pair of thresholds differs from the thresholded surrogate losses of related work, which either penalize all violated thresholds or penalize only the thresholds adjacent to the actual class label.

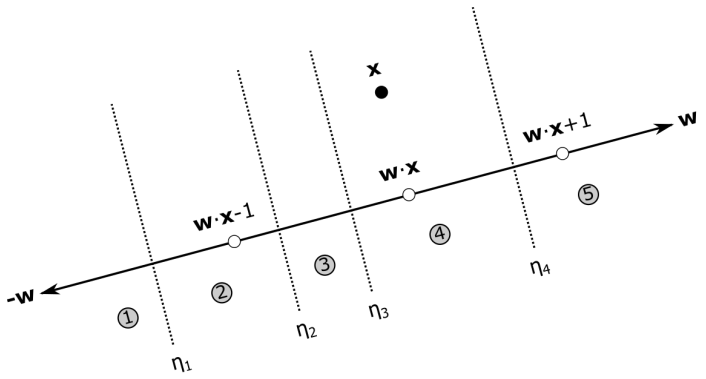


Figure 6: Surrogate loss calculation for datapoint $\mathbf{x}$ (projected to $\mathbf{w}\cdot\mathbf{x}$) with a label prediction of 4 for predictive purposes, the surrogate loss is instead obtained using potentials for the classes based on $\mathbf{w}\cdot\mathbf{x} - 1$ (label 2) and $\mathbf{w}\cdot\mathbf{x} + 1$ (label 5) averaged together.

Using a binary search procedure over $\eta_1, \ldots, \eta_{k-1}$, the largest lower bounding threshold for each of these potentials can be obtained in $\mathcal{O}(\log k)$ time.

#### 3.2.3 Multiclass ordinal surrogate loss

In the multiclass feature representation, we have a set of feature weights $\mathbf{w}_i$ for each label and the adversarial multiclass ordinal surrogate loss can be written as:

$$\mathrm{AL}^{\text{ord-mc}}(\mathbf{x}, y) = \max_{i,j \in [k]} \frac{\mathbf{w}_i \cdot \mathbf{x} + \mathbf{w}_j \cdot \mathbf{x} + j - i}{2} - \mathbf{w}_y \cdot \mathbf{x}.$$

We can also view this as the maximization over $k(k+1)/2$ linear hyperplanes. For an ordinal regression problem with three classes, the loss has six facets with different shapes for each true label value, as shown in Figure 7. In contrast with $\mathrm{AL}^{\text{ord-th}}$, the class potentials for $\mathrm{AL}^{\text{ord-mc}}$ may differ from one another in more-or-less arbitrary ways. Thus, searching for the maximal $i$ and $j$ class labels requires $\mathcal{O}(k)$ time.

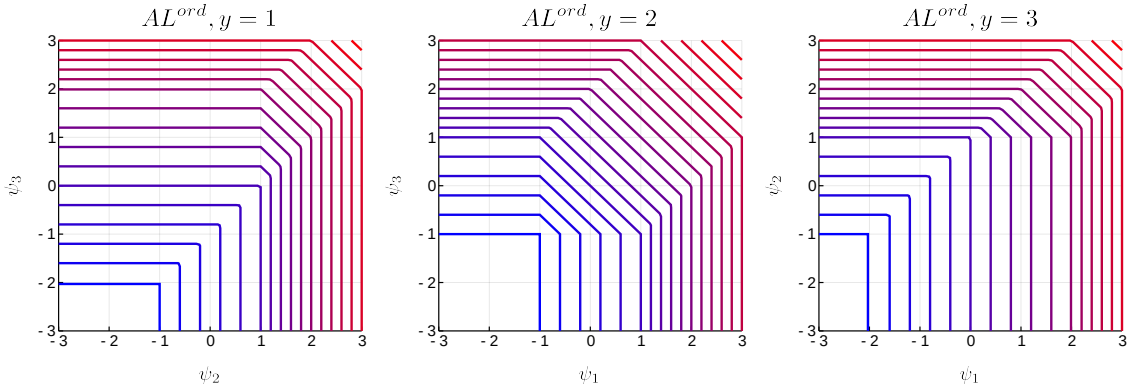


Figure 7: Loss function contour plots of $\mathrm{AL}^{\text{ord}}$ over the space of potential differences $\psi_j \triangleq f_j - f_y$ for the prediction task with three classes when the true label is $y = 1$ (a), $y = 2$ (b), and $y = 3$ (c).

### 3.3 Ordinal Classification with Squared Loss

In some prediction tasks, the squared loss is the preferred metric for ordinal classification to enforce larger penalty as the difference between the predicted and true label increases (Baccianella et al., 2009; Pedregosa et al., 2017). The loss is calculated using the squared difference between label prediction ($\hat{y} \in \mathcal{Y}$) and ground truth label ($y \in \mathcal{Y}$), that is: $\text{loss}(\hat{y}, y) = (\hat{y} - y)^2$. The adversarial surrogate loss for ordinal classification using the squared loss metric is defined in Eq. (4), where $\mathbf{L}$ is the squared loss matrix (e.g. Figure 1c for a five classes ordinal classification). The constraints in Eq. (4) form a convex polytope $\mathbb{C}$. Below is an example of the half-space representation of $\mathbb{C}$ for a four-class ordinal

classification problem with squared loss metric.

$$\begin{array}{l} \\ \text{1st block} \\ \\ \\ \\ \text{2nd block} \\ \\ \\ \\ \text{3rd block} \\ \\ \end{array} \left[\begin{array}{ccccc} 0 & 1 & 4 & 9 & -1 \\ 1 & 0 & 1 & 4 & -1 \\ 4 & 1 & 0 & 1 & -1 \\ 9 & 4 & 1 & 0 & -1 \\ \hdashline 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \hdashline 1 & 1 & 1 & 1 & 0 \\ -1 & -1 & -1 & -1 & 0 \end{array}\right] \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ v \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ -1 \end{bmatrix}.$$

We define the adversarial surrogate loss for ordinal classification with squared loss $\text{AL}^{\text{sq}}$ as stated in Theorem 7.

**Theorem 7** *An adversarial ordinal classification predictor with squared loss is obtained by choosing parameters $\theta$ that minimize the empirical risk of the surrogate loss function:*

$$AL^{sq}(\mathbf{f}, y) = \max \left\{ \max_{\substack{i,j,l \in [k] \\ i<l\leq j}} \frac{(2(j-l)+1)\left[f_i+(l-i)^2\right]+(2(l-i)-1)\left[f_j+(j-l)^2\right]}{2(j-i)},\ \max_i f_i \right\} - f_y.$$

**Proof** The $\text{AL}^{\text{sq}}$ above corresponds to the set of extreme points

$$D = \left\{ \begin{bmatrix} \mathbf{q} \\ v \end{bmatrix} = \tfrac{2(j-l)+1}{2(j-i)} \begin{bmatrix} \mathbf{e}_i \\ (l-i)^2 \end{bmatrix} + \tfrac{2(l-i)-1}{2(j-i)} \begin{bmatrix} \mathbf{e}_j \\ (j-l)^2 \end{bmatrix} \;\middle|\; \begin{array}{c} i,j,l \in [k] \\ i < l \leq j \end{array} \right\} \cup \left\{ \begin{bmatrix} \mathbf{q} \\ v \end{bmatrix} = \begin{bmatrix} \mathbf{e}_i \\ 0 \end{bmatrix} \;\middle|\; i \in [k] \right\}.$$

This means $\mathbf{q}$ can either have one non-zero element with a probability of one or two non-zero elements with the probability specified above.

Similar to the proof of Theorem 5, we next prove that $D \supseteq \text{ext}\ \mathbb{C} = \{\mathbf{c} \in \mathbb{C} : \text{rank}(\mathbf{A_c}) = k+1\}$, as $D \subseteq \mathbb{C}$ is again obvious. Given $\mathbf{c} \in \text{ext}\ \mathbb{C}$, suppose the set of rows that $\mathbf{A_c}$ selected from the first and second block of $\mathbf{A}$ are $S$ and $T$, respectively. Both $S$ and $T$ are subsets of $[k]$, indexed against $\mathbf{A}$. We also denote the set of remaining indices as $R = [k] \backslash T$.

In the case of the squared loss metric, we observe that every row in the first block of $\mathbf{A}$ can be written as a linear combination of two other rows in the first block and the sum-to-one row from the third block. This follows the corresponding relation in continuous squared functions:

$$(x-a)^2 = x^2 - 2ax - a^2 = \alpha(x^2 - 2bx + b^2) + \beta(x^2 - 2cx + c^2) + \gamma = \alpha(x-b)^2 + \beta(x-c)^2 + \gamma,$$

for some value of $\alpha, \beta$, and $\gamma$. Therefore, $S$ can only include one or two elements. This means that $R$ must also contain one or two elements. We consider these two cases:

1. $S$ contains a single element $\{i\}$.
   In this case, $R$ must also be $\{i\}$. If $R = \{j\}$ where $j \neq i$, the equation subsystem requires $v = \mathbf{L}_{(i,:)}\mathbf{q} = (i-j)^2 \geq 1$, since by definition of $R$, $q_j = 1$ and $q_l = 0$ for all

$l \in [k] \backslash j$. However, this contradicts with the requirement of the $j$-th row of $\mathbf{A}$ that $v \leq \mathbf{L}_{(j,:)}\mathbf{q} = 0$. Finally, it is easy to check that the vector in $\mathbb{R}^{k+1}$ that meet all the equalities encoded in this $\mathbf{A_c}$ is $\mathbf{c} = \begin{bmatrix} \mathbf{e}_i \\ 0 \end{bmatrix}$. Obviously $\mathbf{c} \in D$.

2. $S$ contains two elements.
The rank condition requires that $R$ must also contains two elements $\{i, j\}$. Consider these following sub-cases:

   a) $S = \{l-1, l\}$, where $i < l \leq j$.
   Let $\mathbf{c} = \begin{bmatrix} \mathbf{q} \\ v \end{bmatrix}$ be the solution of the equalities encoded in this $\mathbf{A_c}$. By definition of $R$, $q_l = 0$ for all $q \in [k] \backslash \{i, j\}$. The value of $q_i$ and $q_j$ can be calculated by solving $\mathbf{L}_{(l-1,:)}\mathbf{q} = \mathbf{L}_{(l,:)}\mathbf{q}$ or equivalently $\mathbf{L}_{(l-1,i)}q_i + \mathbf{L}_{(l-1,j)}q_j = \mathbf{L}_{(l,i)}q_i + \mathbf{L}_{(l,j)}q_j$, with the constraint that $q_i + q_j = 1$ and the non-negativity constraints. Solving for this equation resulting in the following $q_i$, $q_j$, and $v$:

$$
\begin{aligned}
q_i &= \frac{\mathbf{L}_{(l-1,j)} - \mathbf{L}_{(l,j)}}{\mathbf{L}_{(l,i)} - \mathbf{L}_{(l-1,i)} + \mathbf{L}_{(l-1,j)} - \mathbf{L}_{(l,j)}} \\
&= \frac{(j-l+1)^2 - (j-l)^2}{(l-i)^2 - (l-1-i)^2 + (j-l+1)^2 - (j-l)^2} = \frac{2(j-l)+1}{2\,(j-i)}, \\
q_j &= \frac{\mathbf{L}_{(l,i)} - \mathbf{L}_{(l-1,i)}}{\mathbf{L}_{(l,i)} - \mathbf{L}_{(l-1,i)} + \mathbf{L}_{(l-1,j)} - \mathbf{L}_{(l,j)}} \\
&= \frac{(l-i)^2 - (l-1-i)^2}{(l-i)^2 - (l-1-i)^2 + (j-l+1)^2 - (j-l)^2} = \frac{2(l-i)-1}{2\,(j-i)}, \\
v &= \frac{\left(\mathbf{L}_{(l-1,j)} - \mathbf{L}_{(l,j)}\right)\mathbf{L}_{(l,i)} + \left(\mathbf{L}_{(l,i)} - \mathbf{L}_{(l-1,i)}\right)\mathbf{L}_{(l,j)}}{\mathbf{L}_{(l,i)} - \mathbf{L}_{(l-1,i)} + \mathbf{L}_{(l-1,j)} - \mathbf{L}_{(l,j)}} \\
&= \frac{(2(j-l)+1)\,(l-i)^2 + (2(l-i)-1)\,(j-l)^2}{2\,(j-i)}.
\end{aligned}
$$

   It is obvious that $\mathbf{c} \in D$.

   b) $S = \{m, l\}$, where $i \leq m < l \leq j$ and $m \neq l-1$.
   We want to show that this case is impossible. Solving for the $m$-th and the $l$-th equality, $v = \mathbf{L}_{(m,i)}q_i + \mathbf{L}_{(m,j)}q_j = \mathbf{L}_{(l,i)}q_i + \mathbf{L}_{(l,j)}q_j$ resulting in $q_i = \frac{1}{z}[(j-m)^2 - (j-l)^2]$, $q_j = \frac{1}{z}[(l-i)^2 - (m-i)^2]$, and

$$v = \frac{1}{z}\left\{(l-i)^2[(j-m)^2 - (j-l)^2] + (j-l)^2[(l-i)^2 - (m-i)^2]\right\},$$

   where $z = [(j-m)^2 - (j-l)^2] + [(l-i)^2 - (m-i)^2]$.
   Let $o$ be an index such that $m < o < l$. This row must exist since $m \neq l-1$ and $m < l$. Applying the solution above to the $o$-th row, we define:

$$w \triangleq \mathbf{L}_{(o,:)}\mathbf{q} = \frac{1}{z}\left\{(o-i)^2[(j-m)^2 - (j-l)^2] + (j-o)^2[(l-i)^2 - (m-i)^2]\right\}.$$

   Then,

$$
\begin{aligned}
v - w = \frac{1}{z}\big\{&[(l-i)^2 - (o-i)^2][(j-m)^2 - (j-l)^2] \\
&- [(j-o)^2 - (j-l)^2][(l-i)^2 - (m-i)^2]\big\}.
\end{aligned}
$$

This means that $v - w > 0$, since for all $i \leq m < o < l \leq j$, $i, j, l, m, o \in [k]$,

$$\frac{(l-i)^2 - (o-i)^2}{(l-i)^2 - (m-i)^2} > \frac{(j-o)^2 - (j-l)^2}{(j-m)^2 - (j-l)^2}.$$

Thus, it contradicts with the requirement that $v \leq \mathbf{L}_{(o,:)}$.

c) $S = \{m, l\}$, where $m < i$ or $l > j$.
We first show that $m < i$ is impossible. Note that for $m < i$, the loss value $\mathbf{L}_{(m,i)} = (i-m)^2 > \mathbf{L}_{(i,i)} = 0$ and $\mathbf{L}_{(m,j)} = (j-m)^2 > \mathbf{L}_{(i,j)} = (j-i)^2$. Noting that at least one of $q_i$ or $q_j$ must be positive due to sum-to-one constraint, we conclude that $\mathbf{L}_{(m,:)}\mathbf{q} > \mathbf{L}_{(i,:)}\mathbf{q}$. But this contradicts with $\mathbf{L}_{(m,:)}\mathbf{q} = v \leq \mathbf{L}_{(i,:)}\mathbf{q}$ since the $m \in S$. Similarly, $l > j$ is also impossible.

Therefore, in all possible cases, we have shown that any $\mathbf{c}$ in ext $\mathbb{C}$ must be in $D$, which concludes our proof. ■

Note that $\text{AL}^{\text{sq}}$ contains two separate maximizations corresponding to the case where there are two non-zero elements of $\mathbf{q}$ and the case where only a single non-zero element of $\mathbf{q}$ is possible. Unlike the surrogate for absolute loss, the maximization in $\text{AL}^{\text{sq}}$ cannot be realized independently. A $\mathcal{O}(k^3)$ algorithm is needed to compute the maximization for the case that two non-zero elements of $\mathbf{q}$ are allowed, and a $\mathcal{O}(k)$ algorithm is needed to find the maximum potential in the case of a single non-zero element of $\mathbf{q}$. Therefore, the total runtime of the algorithm for computing $\text{AL}^{\text{sq}}$ is $\mathcal{O}(k^3)$. The loss surface of $\text{AL}^{\text{sq}}$ for the three classes classification is shown in Figure 8.

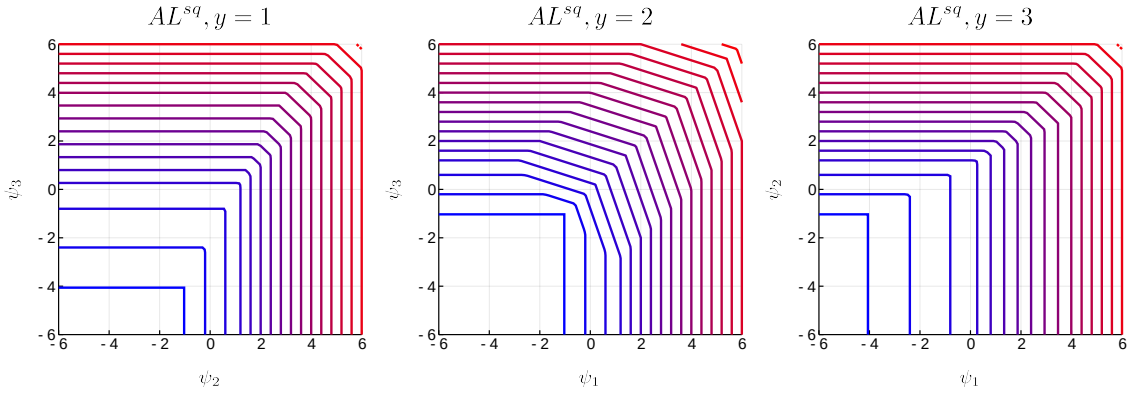


Figure 8: Loss function contour plots of $\text{AL}^{\text{sq}}$ over the space of potential differences $\psi_j \triangleq f_j - f_y$ for the prediction task with three classes when the true label is $y = 1$ (a), $y = 2$ (b), and $y = 3$ (c).

### 3.4 Weighted Multiclass Loss

In more general prediction tasks, the penalty metric for each sample may be different. For example, the predictor may need to prioritize samples with a particular characteristic. In

this subsection, we study the adversarial surrogate loss for weighted multiclass loss, and in particular, the setting with a standard loss metrics weighted by parameter $\alpha$ (for example, the weighted absolute loss: $\text{loss}(\hat{y}, y) = \alpha|\hat{y}-y|$). We next analyze in Theorem 8 the extreme points of the polytope formed by the the constraints in Eq. (4) when $\mathbf{L}$ is the weighted multiclass loss metric.

**Theorem 8** *Let $\mathbf{q}^*$, and $v^*$ be the solution of the adversarial maximin (Eq. (4)) with $\mathbf{L}$ as the loss matrix, then if the loss matrix is $\alpha\mathbf{L}$, the solution of (Eq. (4)) is $\mathbf{q}^\diamond = \mathbf{q}^*$, $v^\diamond = \alpha v^*$.*

**Proof** Multiplying both sides of the constraints $\mathbf{L}_{(i,:)}\mathbf{q} \geq v$ in Eq. (4) and employing $\alpha\mathbf{L}_{(i,:)}\mathbf{q} \geq \alpha v$, we arrive at an equivalent LP problem with the same solution. Therefore, if we replace the original loss metric with $\alpha\mathbf{L}$, then the solution for $\mathbf{q}$ remain the same, and the optimum slack variable value is $\alpha v^*$. ■

Using Theorem 8, we can derive the adversarial surrogate loss for weighted multiclass zero-one loss, absolute loss, and squared loss metrics as stated below.

**Corollary 9** *An adversarial multiclass predictor with weighted zero-one loss is obtained by choosing the parameter $\theta$ that minimizes the empirical risk of the surrogate loss function:*

$$AL^{0\text{-}1\text{-}w}(\mathbf{f}, y, \alpha) = \max_{S\subseteq[k],\, S\neq\emptyset} \frac{\sum_{i\in S} f_i + \alpha\left(|S|-1\right)}{|S|} - f_y.$$

**Corollary 10** *An adversarial ordinal classification predictor with weighted absolute loss is obtained by choosing the parameter $\theta$ that minimizes the empirical risk of the surrogate loss function:*

$$AL^{ord\text{-}w}(\mathbf{f}, y, \alpha) = \max_{i,j\in[k]} \frac{f_i + f_j + \alpha\left(j-i\right)}{2} - f_y.$$

**Corollary 11** *An adversarial ordinal classification predictor with weighted squared loss is obtained by choosing the parameter $\theta$ that minimizes the empirical risk of the surrogate loss function:*

$$AL^{sq\text{-}w}(\mathbf{f}, y, \alpha) = \max\left\{ \max_{\substack{i,j,l\in[k]\\ i<l\leq j}} \frac{(2(j-l)+1)\left[f_i+\alpha(l-i)^2\right]+(2(l-i)-1)\left[f_j+\alpha(j-l)^2\right]}{2(j-i)},\ \max_i f_i \right\} - f_y.$$

The computational cost of calculating the adversarial surrogates for weighted multiclass loss metric above is the same as that for the non-weighted counterpart of the loss, i.e., $\mathcal{O}(k\log k)$ for AL$^{\text{0-1-w}}$, $\mathcal{O}(k)$ for AL$^{\text{ord-w}}$, and $\mathcal{O}(k^3)$ for AL$^{\text{sq-w}}$. The weight constant $\alpha$ does not change the runtime complexity.

### 3.5 Classification with Abstention

In some prediction tasks, it might be better for the predictor to abstain without making any prediction rather than making a prediction with high uncertainty for borderline samples. Under this setting, the standard zero-one loss is used for the evaluation metric with the

addition that the predictor can choose an abstain option and suffer a penalty of $\alpha$. The adversarial surrogate loss for classification with abstention is defined in Eq. (4), where $\mathbf{L}$ is the *abstain* loss matrix (e.g. Figure 1d for a five-class classification with $\alpha = \frac{1}{2}$). The constraints in Eq. (4) form a convex polytope $\mathbb{C}$. Below is the example of the half-space representation of the polytope for a four-class classification problem with abstention.

$$\begin{array}{l} \\ \text{1st block} \\ \\ \\ \\ \hdashline \\ \\ \\ \text{2nd block} \\ \\ \hdashline \\ \text{3rd block} \end{array} \left[\begin{array}{ccccc} 0 & 1 & 1 & 1 & -1 \\ 1 & 0 & 1 & 1 & -1 \\ 1 & 1 & 0 & 1 & -1 \\ 1 & 1 & 1 & 0 & -1 \\ \alpha & \alpha & \alpha & \alpha & -1 \\ \hdashline 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \hdashline 1 & 1 & 1 & 1 & 0 \\ -1 & -1 & -1 & -1 & 0 \end{array}\right] \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ v \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ -1 \end{bmatrix}.$$

Note that the first block of the coefficient matrix $\mathbf{A}$ has $k+1$ rows (one additional row for the abstain option).

We design a convex surrogate loss that can be generalized to the case where $0 \leq \alpha \leq \frac{1}{2}$. We define the adversarial surrogate loss for classification with abstention $\text{AL}^{\text{abstain}}$ as stated in Theorem 12 below.

**Theorem 12** *An adversarial predictor for classification with abstention with the penalty for abstain option is $\alpha$ where $0 \leq \alpha \leq \frac{1}{2}$, is obtained by choosing the parameter $\theta$ that minimizes the empirical risk of the surrogate loss function:*

$$AL^{abstain}(\mathbf{f}, y, \alpha) = \max\left\{ \max_{i,j \in [k], i \neq j} (1-\alpha) f_i + \alpha f_j + \alpha,\ \max_i f_i \right\} - f_y.$$

**Proof** The $\text{AL}^{\text{abstain}}$ above corresponds to the set of extreme points

$$D = \left\{ \begin{bmatrix} \mathbf{q} \\ v \end{bmatrix} = (1-\alpha) \begin{bmatrix} \mathbf{e}_i \\ 0 \end{bmatrix} + \alpha \begin{bmatrix} \mathbf{e}_j \\ 1 \end{bmatrix} \;\middle|\; \begin{array}{c} i, j \in [k] \\ i \neq j \end{array} \right\} \cup \left\{ \begin{bmatrix} \mathbf{q} \\ v \end{bmatrix} = \begin{bmatrix} \mathbf{e}_i \\ 0 \end{bmatrix} \;\middle|\; i \in [k] \right\}.$$

This means $\mathbf{q}$ can only have one non-zero element with probability of one or two non-zero elements with the probability of $\alpha$ and $(1-\alpha)$.

Similar to the proof of Theorem 5, we next prove that $D \supseteq \text{ext}\,\mathbb{C} = \{\mathbf{c} \in \mathbb{C} : \text{rank}(\mathbf{A_c}) = k+1\}$, as $D \subseteq \mathbb{C}$ is again obvious. Given $\mathbf{c} \in \text{ext}\,\mathbb{C}$, suppose the set of rows that $\mathbf{A_c}$ selected from the first and second block of $\mathbf{A}$ are $S$ and $T$, respectively. Now $S$ is a subset of $[k+1]$ where the $(k+1)$-th index represents the abstain option, while $T$ is a subset of $[k]$, indexed against $\mathbf{A}$. Similar to the case of zero-one loss metric, $S$ and $T$ must be disjoint. We also denote the set of remaining indices as $R = [k] \backslash T$.

The abstain row in the first block of $\mathbf{A}$ implies that $v \leq \alpha$, while including $j$ regular rows to $S$ implies that $v = \frac{j-1}{j}$. Therefore, only a single regular row can be in $S$ when $\alpha < \frac{1}{2}$ or at most two regular rows can be in $S$ when $\alpha = \frac{1}{2}$.

We first consider $\alpha < \frac{1}{2}$. Let $S = \{i, k+1\}$, i.e., one regular row and one abstain row. Due to rank requirement of $\mathbf{A_c}$ and the disjointness of $S$ and $T$, $R$ must contain two elements with one of them be $i$, i.e. $R = \{i, j\}$. To get the value of $q_i$ and $q_j$, we solve for the equation $\mathbf{L}_{(i,:)}\mathbf{q} = \mathbf{L}_{(k+1,:)}\mathbf{q}$ which can be simplified as $q_j = \alpha q_i + \alpha q_j$. The solution is to set $q_i = (1-\alpha)$, $q_j = \alpha$, and $v = \alpha$, which obviously in $D$. For the second case, let $S = \{i\}$, i.e., one regular row. In this case $R$ must be $\{i\}$ too. This yields $\mathbf{c}$ with $q_i = 1$, $q_j = 0, \forall j \in [k] \backslash i$, and $v = 0$. Obviously $\mathbf{c} \in D$.

For the case where $\alpha = \frac{1}{2}$, two cases above still apply with two additional cases. First, $S = \{i, j\}$, i.e., two regular rows. In this case, $R$ must be $\{i, j\}$ too. The solution is to set $q_i = q_j = \frac{1}{2}$, and $v = \frac{1}{2}$. This satisfies $v = \mathbf{L}_{(i,:)}\mathbf{q} = \mathbf{L}_{(j,:)}\mathbf{q} = \frac{1}{2}$ as well as $v \leq \mathbf{L}_{(k+1,:)}\mathbf{q} = \alpha = \frac{1}{2}$. Obviously, this is in $D$. Second, $S = \{i, j, k+1\}$, i.e., two regular rows and one abstain row. Due to the rank requirement of $\mathbf{A_c}$, and the disjointness of $S$ and $T$, $R$ must contain three elements:, $i, j$, and another index $l \in [k] \backslash \{i, j\}$. It is easy to check that the solution in this case is also to set $q_i = q_j = \frac{1}{2}$, and $v = \frac{1}{2}$. This satisfies $v = \mathbf{L}_{(i,:)}\mathbf{q} = \mathbf{L}_{(j,:)}\mathbf{q} = \frac{1}{2}$ as well as $v = \mathbf{L}_{(k+1,:)}\mathbf{q} = \alpha = \frac{1}{2}$.

Therefore, in all possible cases, we have shown that any $\mathbf{c}$ in ext $\mathbb{C}$ must be in $D$. ■

We can view the maximization in $\text{AL}^{\text{abstain}}$ as the maximization over $k^2$ linear hyperplanes, with $k$ hyperplanes are defined by the case where only a single element of $\mathbf{q}$ can be non zero and the rest $k(k-1)$ hyperplanes are defined by the case where two elements of $\mathbf{q}$ are non zero. For the binary classification with abstention problem, the surrogate loss function has four facets. Figure 9 shows the loss function in the case where $\alpha = \frac{1}{3}$ and $\alpha = \frac{1}{2}$. Note that for $\alpha = \frac{1}{2}$ the facet corresponds with the hyperplane of $(1-\alpha)f_1 + \alpha f_2 + \alpha$ collide with the facet corresponds with the hyperplane of $(1-\alpha)f_2 + \alpha f_1 + \alpha$, resulting in a loss function with only three facets. For the three-class classification with abstention problem, the surrogate loss has nine facets with different shapes for each true label value, as shown in Figure 10 for $\alpha = \frac{1}{3}$ and $\alpha = \frac{1}{2}$. Similar to the binary classification case, for $\alpha = \frac{1}{2}$, some facets in the surrogate loss surface collide resulting in a surrogate loss function with only six facets.

Even though the maximization in $\text{AL}^{\text{abstain}}$ is over $n^2$ different items, we construct a faster algorithm to compute the loss. The algorithm keeps track of the two largest potentials as it scans all $k$ potentials. Denote $i^*$ and $j^*$ as the index of the best and the second best potentials respectively. The algorithm then takes the maximum of two candidate solutions: (1) assigning all the probability to $f_{i*}$, resulting in the loss value of $f_{i*}$, or (2) assigning $1-\alpha$ probability to $f_{i*}$ and $\alpha$ probability to $f_{j*}$, resulting in the loss value of $(1-\alpha)f_{i^*} + \alpha f_{j^*} + \alpha$. The runtime of this algorithm is $\mathcal{O}(k)$ due to the need to scan all $k$ potentials once.

### 3.6 General Multiclass Loss

For a general multiclass loss matrix $\mathbf{L}$, the extreme points of the polytope defined by the constraints in Eq. (4) may not be easily characterized. Nevertheless, since the maximization in Eq. (4) is in the form of a linear program (LP), some well-known algorithms for linear programming can be used to solve the problem. The techniques for solving LPs have been extensively studied, resulting in two major algorithms:

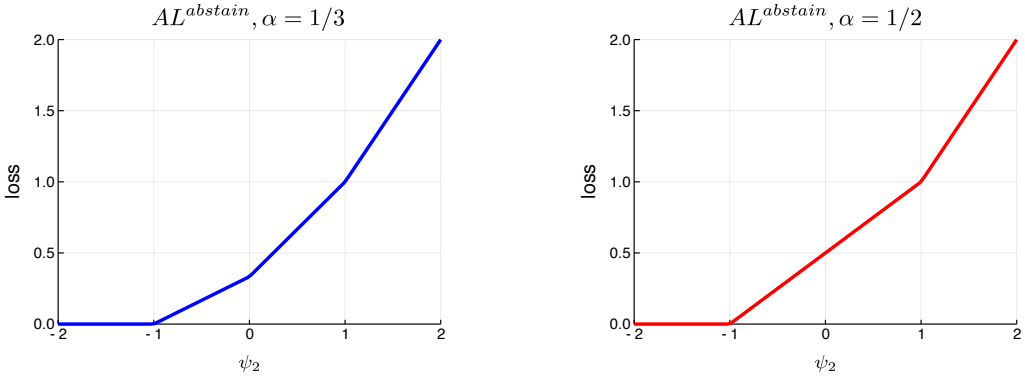


Figure 9: $\text{AL}^{\text{abstain}}$ evaluated over the space of potential differences ($\psi_{i,y} = f_i - f_y$; and $\psi_{i,i} = 0$) for binary prediction tasks when the true label is $y = 1$, where $\alpha = \frac{1}{3}$ (a), and $\alpha = \frac{1}{2}$ (b).

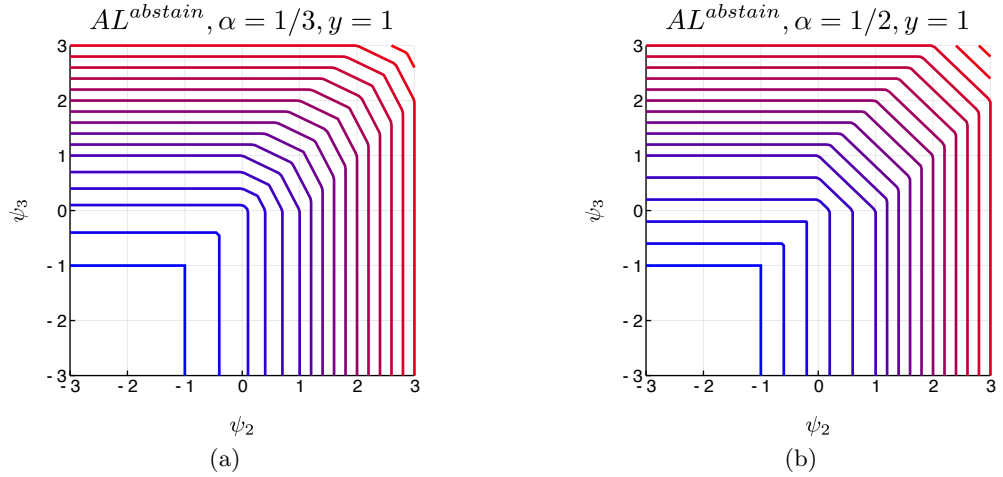


Figure 10: Loss function contour plots of $\text{AL}^{\text{abstain}}$ over the space of potential differences $\psi_j \triangleq f_j - f_y$ for the prediction task with three classes when the true label is $y = 1$, where $\alpha = \frac{1}{3}$ (a), and $\alpha = \frac{1}{2}$ (b).

1. Simplex algorithm.
   The simplex algorithm (Dantzig, 1948, 1963) cleverly visits the extreme points in the convex polytope until it reaches the one that maximizes the objective. This is the most popular algorithm for solving LP problems. However, although the algorithm typically works well in practice, the worst case complexity of the algorithm is exponential in the problem size.

2. Interior point algorithm.
   Karmarkar (1984) proposed an interior point algorithm for solving LPs with polynomial worst case runtime complexity. The algorithm finds the optimal solution by traversing the interior of the feasible region. The runtime complexity of Karmarkar's algorithm for solving the LP is $\mathcal{O}(n^{3.5})$ where $n$ is the number of variables in the LP problem. In Eq. (4), $n = k + 1$.

Therefore, using Karmarkar's algorithm we can bound the worst-case runtime complexity of computing the adversarial surrogate for arbitrary loss matrix $\mathbf{L}$ with $\mathcal{O}(k^{3.5})$ where $k$ is the number of classes.

## 4. Prediction Formulation

The dual formulation of the adversarial prediction (Eq. (2)) provides a way to construct a learning algorithm for the framework. The learning step in the adversarial prediction is to find the optimal Lagrange dual variable $\theta^* = \min_\theta \mathbb{E}_{\mathbf{X},Y\sim\tilde{P}} [AL(\mathbf{X}, Y, \theta)]$. In the prediction step, we use the optimal $\theta^*$ to make a label prediction given newly observed data. Although $\theta^*$ is only optimized with respect to the conditional probability at the data points $\mathbf{x}$ in the training set, we assume that it can be generalized to the true data generating distribution, including the newly observed data points in the testing set.

### 4.1 Probabilistic Prediction

Given a new data point $\mathbf{x}$ and its label $y$, and the optimal $\theta^*$, we formulate the prediction minimax game based on Eq. (2) by flipping the optimization order between the predictor and the adversary player:

$$\min_{\hat{P}(\hat{Y}|\mathbf{x})} \max_{\check{P}(\check{Y}|\mathbf{x})} \mathbb{E}_{\hat{Y}|\mathbf{x}\sim\hat{P};\check{Y}|\mathbf{x}\sim\check{P}} \left[\text{loss}(\hat{Y}, \check{Y}) + \theta^{*\intercal} \left(\phi(\mathbf{x}, \check{Y}) - \phi(\mathbf{x}, y)\right)\right].$$

This flipping is enabled by the strong minimax duality theorem (Von Neumann and Morgenstern, 1945). Denoting $f_i = \theta^{*\intercal}\phi(\mathbf{x}, i)$, the prediction formulation can be written in our vector and matrix notation as:

$$\min_{\mathbf{p}\in\Delta} \max_{\mathbf{q}\in\Delta} \mathbf{p}^\intercal \mathbf{L}\mathbf{q} + \mathbf{f}^\intercal \mathbf{q} - f_y. \tag{8}$$

Even though the ground truth label $y$ serves an important role in the learning step (Eq. (2)), it is constant with respect to the predictor probability $\mathbf{p}$. Therefore, to get the optimal prediction probability $\mathbf{p}^*$, the term $f_y$ in Eq. (8) can be removed, resulting in the following probabilistic prediction formulation:

$$\mathbf{p}^* = \operatorname*{argmin}_{\mathbf{p}\in\Delta} \max_{\mathbf{q}\in\Delta} \mathbf{p}^\intercal \mathbf{L}\mathbf{q} + \mathbf{f}^\intercal \mathbf{q}. \tag{9}$$

### 4.2 Non-probabilistic Prediction

In some prediction tasks, a learning algorithm needs to provide a single class label prediction rather than a probabilistic prediction. We propose two prediction schemes to get a non-probabilistic single label prediction $y^*$ from our formulation.

1. The maximizer of the potential $\mathbf{f}$.
   This follows the standard prediction technique used by many ERM-based models, e.g., SVM. Given the best parameter $\theta^*$, the predicted label is computed by choosing the label that maximizes the potential value, i.e.,

$$y^* = \operatorname*{argmax}_i f_i, \qquad \text{where: } f_i = \theta^{*\intercal}\phi(\mathbf{x}, i).$$

   Note that this prediction scheme works for the prediction settings where the predictor employs the same set of class labels as the ground truth, i.e., $y^* \in \mathcal{Y}$ and $y \in \mathcal{Y}$ where $\mathcal{Y} = [k]$. If they are different such as in the classification task with abstention, this prediction scheme cannot be used. The runtime complexity of this prediction scheme is $\mathcal{O}(k)$ for $k$ classes.

2. The maximizer of the predictor's optimal probability $\mathbf{p}^*$.
   This prediction scheme requires the predictor to first produce a probabilistic prediction by using Eq. (9). Then the algorithm chooses the label that maximizes the conditional probability, i.e.,

$$y^* = \operatorname*{argmax}_i p_i^*, \qquad \text{where: } \mathbf{p}^* = \operatorname*{argmin}_{\mathbf{p}\in\Delta} \max_{\mathbf{q}\in\Delta} \mathbf{p}^\intercal \mathbf{L}\mathbf{q} + \mathbf{f}^\intercal \mathbf{q}.$$

   This prediction scheme can be applied to more general problems, including the case where the predictor and ground truth class labels are chosen from different sets of labels. This is useful for the classification task with abstention. However, for a general loss matrix $\mathbf{L}$, this prediction scheme is more computation intensive than the potential-based prediction, i.e., $\mathcal{O}(k^{3.5})$ due to the need of solving the minimax game by linear programming (Karmarkar's algorithm).

### 4.3 Prediction Algorithm for Classification with Abstention

In the task of classification with abstention, the standard prediction scheme using the potential maximizer $\operatorname{argmax}_i f_i$ cannot be applied due to the additional abstain option of the predictor. In this subsection, we construct a fast prediction scheme that is based on the predictor's optimal probability in the minimax game (Eq. (9)) without the need to use general purpose LP solver. The minimax game in Eq. (9) can be equivalently written in the standard LP form as:

$$\begin{aligned} &\min_{\mathbf{p},v} v \\ &\text{s.t.: } v \geq \mathbf{L}_{(:,i)}{}^\intercal \mathbf{p} + f_i, \quad \forall i \in [k] \\ &\qquad \mathbf{p} \in \mathbb{R}_+^{k+1}, \\ &\qquad \mathbf{p}^\intercal \mathbf{1} = 1, \end{aligned} \tag{10}$$

where $v$ is a slack variable to convert the inner maximization into linear constraints, and $\mathbf{L}_{(:,i)}$ denotes the $i$-th column of the loss matrix $\mathbf{L}$. We aim to analyze the optimal $\mathbf{p}$ and $v$

for the case where $\mathbf{L}$ is the loss matrix for classification with abstention, e.g.,

$$\mathbf{L} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ \alpha & \alpha & \alpha & \alpha \end{bmatrix}$$

in a four-class classification, where $\alpha$ is the penalty for abstaining (c.f. Section 3.5). Similar to the case of the adversarial surrogate loss for classification with abstention, our analysis can be generalized to the case where $0 \leq \alpha \leq \frac{1}{2}$.

**Theorem 13** *Let $\alpha$ be the penalty for abstaining where $0 \leq \alpha \leq \frac{1}{2}$, $\theta^*$ be the learned parameter, and $\mathbf{f}$ be the potential vector for all classes where $f_i = \theta^{*\intercal}\phi(\mathbf{x}, i)$. Given a new data point $\mathbf{x}$, let $i^* = \operatorname{argmax}_i f_i$ (break tie arbitrarily), $j^* = \operatorname{argmax}_{j \neq i^*} f_j$, and $\mathbf{e}_{i^*} \in \mathbb{R}^k$ be the $i^*$-th cannonical vector. Then the predictor's optimal probability $\mathbf{p}^*$ of Eq. (10) for the task of classification with abstention can be directly computed as:*

$$\mathbf{p}^* = \begin{bmatrix} \mathbf{e}_{i^*} \\ 0 \end{bmatrix} \text{ if } f_{i^*} - f_{j^*} \geq 1 \qquad \text{and} \qquad \mathbf{p}^* = \begin{bmatrix} (f_{i^*} - f_{j^*})\mathbf{e}_{i^*} \\ 1 - f_{i^*} + f_{j^*} \end{bmatrix} \text{ if } f_{i^*} - f_{j^*} < 1.$$

**Proof** Based on Theorem 12, the optimal objective value of (10) is exactly $\mathrm{AL}^{\mathrm{abstain}}(\mathbf{f}, y, \alpha) + f_y$, which is $f_{i^*}$ when $f_{i^*} - f_{j^*} \geq 1$, and $\alpha + (1-\alpha)f_{i^*} + \alpha f_{j^*}$ otherwise. So we only need to verify that the $\mathbf{p}^*$ given in the theorem attains these two values, or equivalently, $\max_i \left\{\mathbf{L}_{(:,i)}^{\intercal}\mathbf{p}^* + f_i\right\}$ attains these two values.

1. Case 1: $f_{i^*} - f_{j^*} \geq 1$. Now $\mathbf{p}^* = \begin{bmatrix} \mathbf{e}_{i^*} \\ 0 \end{bmatrix}$ renders $\mathbf{L}_{(:,i^*)}^{\intercal}\mathbf{p}^* + f_{i^*} = f_{i^*}$, and $\mathbf{L}_{(:,k)}^{\intercal}\mathbf{p}^* + f_k = 1 + f_k \leq f_{i^*}$ for all $k \neq i^*$. So the objective of (10) matches $\mathrm{AL}^{\mathrm{abstain}} + f_y$.

2. Case 2: $f_{i^*} - f_{j^*} < 1$. Now $\mathbf{p}^* = \begin{bmatrix} (f_{i^*} - f_{j^*})\mathbf{e}_{i^*} \\ 1 - f_{i^*} + f_{j^*} \end{bmatrix} \in \mathbb{R}_+^{k+1}$ and $\mathbf{1}^{\intercal}\mathbf{p}^* = 1$. Furthermore,

   $$\mathbf{L}_{(:,i^*)}^{\intercal}\mathbf{p}^* + f_{i^*} = \alpha(1 - f_{i^*} + f_{j^*}) + f_{i^*},$$
   $$\mathbf{L}_{(:,k)}^{\intercal}\mathbf{p}^* + f_k = f_{i^*} - f_{j^*} + \alpha(1 - f_{i^*} + f_{j^*}) + f_k \leq \alpha(1 - f_{i^*} + f_{j^*}) + f_{i^*} \quad (k \neq i^*).$$

   Therefore $\max_i \left\{\mathbf{L}_{(:,i)}^{\intercal}\mathbf{p}^* + f_i\right\} = \alpha(1 - f_{i^*} + f_{j^*}) + f_{i^*}$, which matches $\mathrm{AL}^{\mathrm{abstain}} + f_y$.

■

From the theorem above, we derive a non-probabilistic prediction scheme based on the maximizer of the predictor's probability as follows.

**Corollary 14** *For $0 \leq \alpha \leq \frac{1}{2}$, a non-probabilistic prediction of the adversarial prediction method for the classification with abstention task can be computed as:*

$$y^* = \begin{cases} i^* & f_{i^*} - f_{j^*} \geq \frac{1}{2} \\ \textit{abstain} & \textit{otherwise} \end{cases}$$

*where $i^*$ and $j^*$ are the indices of the largest and the second largest potentials respectively.*

The runtime complexity of this prediction scheme is $\mathcal{O}(k)$ since the algorithm needs to scan all $k$ potentials and maintain the two largest potentials. This is much faster than solving the minimax game in Eq. (9), which costs $\mathcal{O}(k^{3.5})$.

## 5. Fisher Consistency

The behavior of a prediction method in ideal learning settings—i.e., trained on the true evaluation distribution and given an arbitrarily rich feature representation, or, equivalently, considering the space of all measurable functions—provides a useful theoretical validation. Fisher consistency requires that the prediction model yields the Bayes optimal decision boundary in this setting (Tewari and Bartlett, 2007; Liu, 2007; Ramaswamy and Agarwal, 2012; Pedregosa et al., 2017). Suppose the potential scoring function $f(\mathbf{x}, y)$ is optimized over the space of all measurable functions. Given the true distribution $P(\mathbf{X}, Y)$, a surrogate loss function $\delta$ is said to be Fisher consistent with respect to the loss $\ell$ if the minimizer $f^*$ of the surrogate loss reaches the Bayes optimal risk, i.e.:

$$f^* \in \operatorname*{argmin}_f \mathbb{E}_{Y|\mathbf{x}\sim P}\left[\delta_f(\mathbf{x}, Y)\right] \;\Rightarrow\; \mathbb{E}_{Y|\mathbf{x}\sim P}\left[\ell_{f^*}(\mathbf{x}, Y)\right] = \min_f \mathbb{E}_{Y|\mathbf{x}\sim P}\left[\ell_f(\mathbf{x}, Y)\right]. \quad (11)$$

Here $\delta_f(\mathbf{x}, y)$ stands for the surrogate loss function value if the true label is $y$ and we make a prediction on $\mathbf{x}$ using the potential function $f(\mathbf{x}, y)$. The loss $\ell_f$ has a similar meaning.

### 5.1 Fisher Consistency for Potential-Based Prediction

We consider Fisher consistency for standard multiclass classification where the prediction is done by taking the argmax of the potentials, i.e., $\operatorname{argmax}_y f(\mathbf{x}, y)$. This usually applies to the setting where the predictor and ground truth class labels are chosen from the same set of labels, i.e., $y^* \in \mathcal{Y}$, and $y \in \mathcal{Y} \triangleq [k]$. Given that prediction is based on the argmax of the potentials, the right-hand side of Eq. (11) is equivalent to:

$$\mathbb{E}_{Y|\mathbf{x}\sim P}\left[\ell(\operatorname*{argmax}_{y'} f^*(\mathbf{x}, y'), Y)\right] = \min_f \mathbb{E}_{Y|\mathbf{x}\sim P}\left[\ell(\operatorname*{argmax}_{y'} f(\mathbf{x}, y'), Y)\right].$$

Since $f$ is optimized over all measurable functions, the condition in Eq. (11) can be further simplified as

$$\begin{aligned} & f^* \in \operatorname*{argmin}_f \mathbb{E}_{Y|\mathbf{x}\sim P}\left[\delta_f(\mathbf{x}, Y)\right] \\ \Rightarrow\; & \operatorname*{argmax}_{y'} f^*(\mathbf{x}, y') \subseteq \operatorname*{argmin}_{y'} \mathbb{E}_{Y|\mathbf{x}\sim P}\left[\ell(y', Y)\right], \qquad \forall \mathbf{x} \in \mathcal{X}. \end{aligned}$$

Using the potential scoring function notation $f(\mathbf{x}, y)$, the adversarial surrogate loss in Eq. (3) can be equivalently written as:

$$AL_f(\mathbf{x}, y) = \max_{\check{P}(\check{Y}|\mathbf{x})} \min_{\hat{P}(\hat{Y}|\mathbf{x})} \mathbb{E}_{\hat{Y}|\mathbf{x}\sim\hat{P};\check{Y}|\mathbf{x}\sim\check{P}}\left[\text{loss}(\hat{Y}, \check{Y}) + f(\mathbf{x}, \check{Y}) - f(\mathbf{x}, y)\right].$$

Then, the Fisher consistency condition for the adversarial surrogate loss $\mathrm{AL}_f$ becomes:

$$f^* \in \mathcal{F}^* \triangleq \operatorname*{argmin}_f \mathbb{E}_{Y|\mathbf{x}\sim P}\left[\mathrm{AL}_f(\mathbf{x}, Y)\right] \tag{12}$$
$$\Rightarrow \operatorname*{argmax}_y f^*(\mathbf{x}, y) \subseteq \mathcal{Y}^\diamond \triangleq \operatorname*{argmin}_{y'} \mathbb{E}_{Y|\mathbf{x}\sim P}[\mathrm{loss}(y', Y)].$$

In the sequel, we will show that the condition in Eq. (12) holds for our adversarial surrogate AL for any loss metrics satisfying a natural requirement that the correct prediction must suffer a loss that is strictly less than incorrect predictions. We start in Theorem 15 by establishing Fisher consistency when the optimal label is unique (i.e., $\mathcal{Y}^\diamond$ is a singleton), and then proceed to more general cases in Theorem 16.

**Theorem 15** *In the standard multiclass classification setting, suppose we have a loss metric that satisfies the natural requirement: $\mathrm{loss}(y, y) < \mathrm{loss}(y, y')$ for all $y' \neq y$. Then the adversarial surrogate loss $AL_f$ is Fisher consistent if $f$ is optimized over all measurable functions and $\mathcal{Y}^\diamond$ is a singleton.*

**Proof** Let $\mathbf{p}$ be the probability mass given by the predictor player $\hat{P}(\hat{Y}|\mathbf{x})$, $\mathbf{q}$ be the probability mass given by the adversary player $\check{P}(\check{Y}|\mathbf{x})$, and $\mathbf{d}$ be the probability mass of the true distribution $P(Y|\mathbf{x})$. So, all $\mathbf{p}$, $\mathbf{q}$, and $\mathbf{d}$ lie in the $k$ dimensional probability simplex $\Delta$, where $k$ is the number of classes. Let $\mathbf{L}$ be a $k$-by-$k$ loss matrix whose $(y, y')$-th entry is $\mathrm{loss}(y, y')$. Let $\mathbf{f} \in \mathbb{R}^k$ be the vector encoding of the value of $f$ at all classes. The definition of $f^*$ in Eq. (12) now becomes:

$$\mathbf{f}^* \in \operatorname*{argmin}_{\mathbf{f}} \max_{\mathbf{q}\in\Delta} \min_{\mathbf{p}\in\Delta} \{\mathbf{f}^\intercal\mathbf{q} + \mathbf{p}^\intercal\mathbf{L}\mathbf{q} - \mathbf{d}^\intercal\mathbf{f}\} = \operatorname*{argmin}_{\mathbf{f}} \max_{\mathbf{q}\in\Delta} \left\{\mathbf{f}^\intercal\mathbf{q} + \min_y(\mathbf{L}\mathbf{q})_y - \mathbf{d}^\intercal\mathbf{f}\right\}. \tag{13}$$

Since $\mathcal{Y}^\diamond \triangleq \operatorname{argmin}_y \mathbb{E}_{Y|\mathbf{x}\sim P}[\mathrm{loss}(y, Y)]$ (or equivalently $\operatorname{argmin}_y(\mathbf{L}\mathbf{d})_y$) contains only a singleton, we denote it as $y^\diamond$. We are to show that $\operatorname{argmax}_y f^*(\mathbf{x}, y)$ is a singleton, and its only element is exactly $y^\diamond$. Since $\mathbf{f}^*$ is an optimal solution, the objective function must have a zero subgradient at $\mathbf{f}^*$. That means $\mathbf{0} = \mathbf{q}^* - \mathbf{d}$, where $\mathbf{q}^*$ is an optimal solution in Eq. (13) under $\mathbf{f}^*$. As a result:

$$\mathbf{d} \in \operatorname*{argmax}_{\mathbf{q}\in\Delta} \left\{\mathbf{q}^\intercal\mathbf{f}^* + \min_y(\mathbf{L}\mathbf{q})_y\right\}. \tag{14}$$

By the first order optimality condition of constrained convex optimization (see Eq. (4.21) of Boyd and Vandenberghe (2004)), this means:

$$\left(\mathbf{f}^* + \mathbf{L}_{(y^\diamond,:)}{}^\intercal\right)^\intercal (\mathbf{u} - \mathbf{d}) \leq 0 \quad \forall \mathbf{u} \in \Delta, \tag{15}$$

where $\mathbf{L}_{(y^\diamond,:)}$ is the $y^\diamond$-th row of $\mathbf{L}$, $\mathbf{f}^* + \mathbf{L}_{(y^\diamond,:)}{}^\intercal$ is the gradient of the objective in Eq. (14) with respect to $\mathbf{q}$ evaluated at $\mathbf{q} = \mathbf{d}$. Here we used the definition of $y^\diamond$. However, this inequality can hold for some $\mathbf{d} \in \Delta_k \cap \mathbb{R}^k_{++}$ only if $\mathbf{f}^* + \mathbf{L}_{(y^\diamond,:)}{}^\intercal$ is a uniform vector, i.e., $f^*_y + \mathrm{loss}(y^\diamond, y)$ is constant in $y$. To see this, let us assume the contrary that $\mathbf{v} \triangleq \mathbf{f}^* + \mathbf{L}_{(y^\diamond,:)}{}^\intercal$ is not a uniform vector, and let $i$ be the index of its maximum element. Setting $\mathbf{u} = \mathbf{e}_i$, it is

clear that for any $\mathbf{d} \in \Delta_k \cap \mathbb{R}^k_{++}$, $\mathbf{v}^\intercal \mathbf{u} > \mathbf{v}^\intercal \mathbf{d}$ and hence $\left(\mathbf{f}^* + \mathbf{L}_{(y^\diamond,:)}{}^\intercal\right)^\intercal (\mathbf{u} - \mathbf{d}) > 0$, which violates the optimality condition.

Finally, using the assumption that $\text{loss}(y, y) < \text{loss}(y, y')$ for all $y' \neq y$, it follows that $\text{argmax}_y f^*(\mathbf{x}, y) = \text{argmin}_y \mathbf{L}_{(y^\diamond, y)} = \{y^\diamond\}$. ■

The assumption of loss function in the above theorem is quite mild, requiring only that the incorrect predictions suffer higher loss than the correct one. We do not even require symmetry in its two arguments. The key to the proofs is the observation that for the optimal potential function $f^*$, $f^*(\mathbf{x}, y) + \text{loss}(y^\diamond, y)$ is invariant to $y$ when $\mathcal{Y}^\diamond = \{y^\diamond\}$. We refer to this as the *loss reflective* property of the minimizer. In the next theorem, we generalize Theorem 15 to the case where the Bayes optimal prediction may have ties.

**Theorem 16** *In the standard multiclass classification setting, suppose we have a loss metric that satisfies the natural requirement: loss$(y, y) <$ loss$(y, y')$ for all $y' \neq y$. Furthermore, if $f$ is optimized over all measurable functions, then:*

*(a) there exists $f^* \in \mathcal{F}^*$ such that $\text{argmax}_y f^*(\mathbf{x}, y) \subseteq \mathcal{Y}^\diamond$ (i.e., satisfies the Fisher consistency requirement). In fact, all elements in $\mathcal{Y}^\diamond$ can be recovered by some $f^* \in \mathcal{F}^*$.*

*(b) if the loss satisfies $\text{argmin}_{y'} \sum_{y \in \mathcal{Y}^\diamond} \alpha_y \text{loss}(y, y') \subseteq \mathcal{Y}^\diamond$ for all $\alpha_{(\cdot)} \geq 0$ and $\sum_{y \in \mathcal{Y}^\diamond} \alpha_y = 1$, then $\text{argmax}_y f^*(\mathbf{x}, y) \subseteq \mathcal{Y}^\diamond$ for **all** $f^* \in \mathcal{F}^*$. In this case, all $f^* \in \mathcal{F}^*$ satisfies the Fisher consistency requirement.*

**Proof** Let $\mathbf{p}$, $\mathbf{q}$, and $\mathbf{d}$ have the same meaning as in the proof of Theorem 15. Let $\mathcal{Y}^\diamond \triangleq \text{argmin}_y (\mathbf{L}\mathbf{d})_y$ which is not necessarily a singleton. The analysis in the proof of Theorem 15 carries over to this case, except for Eq. (15). Denote $h(\mathbf{q}) \triangleq \mathbf{q}^\intercal \mathbf{f}^* + \min_y (\mathbf{L}\mathbf{q})_y$. The subdifferential of $-h(\mathbf{q})$ evaluated at $\mathbf{q} = \mathbf{d}$ is the set:

$$\partial(-h)(\mathbf{d}) = \{-\mathbf{f}^* - \mathbf{v} \mid \mathbf{v} \in \mathbf{conv}\{\mathbf{L}_{(y^\diamond,:)}{}^\intercal \mid y^\diamond \in \mathcal{Y}^\diamond\}\}, \tag{16}$$

where **conv** denotes the convex hull. By extending the first order optimality condition to the subgradient case, this means that there is a subgradient $\mathbf{g} \in \partial(-h)(\mathbf{d})$ such that:

$$\mathbf{g}^\intercal (\mathbf{u} - \mathbf{d}) \geq 0 \quad \forall \mathbf{u} \in \Delta.$$

Similar to the singleton $\mathcal{Y}^\diamond$ case, this inequality can hold for some $\mathbf{d} \in \Delta \cap \mathbb{R}^k_{++}$ only if $\mathbf{g}$ is a uniform vector. Based on Eq. (16), $-\mathbf{g} - \mathbf{f}^*$ can be written as a convex combination of $\{\mathbf{L}_{(y^\diamond,:)}{}^\intercal \mid y^\diamond \in \mathcal{Y}^\diamond\}$, and the "if and only if" relationship in the above derivation leads to a full characterization of the optimal potential function set $\mathcal{F}^*_{\mathbf{x}}$ for a given $\mathbf{x}$ (c.f. Eq. (12)):

$$\mathcal{F}^*_{\mathbf{x}} = \left\{ \mathbf{f}^* = c\mathbf{1} - \sum_{y \in \mathcal{Y}^\diamond} \alpha_y \mathbf{L}_{(y,:)}{}^\intercal \;\middle|\; \alpha_{(\cdot)} \geq 0,\ \sum_{y \in \mathcal{Y}^\diamond} \alpha_y = 1,\ c \in \mathbb{R} \right\}. \tag{17}$$

This means that multiple solutions of $\mathbf{f}^*$ are possible. For each element $y^\diamond$ in $\mathcal{Y}^\diamond$, we can recover a $f^*_{y^\diamond}$ in which the $\text{argmax}_y f^*_{y^\diamond}(\mathbf{x}, y)$ contains a singleton element $y^\diamond$ by using Eq. (17) with $\alpha_{y^\diamond} = 1$ and $\alpha_{y \in \{\mathcal{Y}^\diamond \setminus y^\diamond\}} = 0$. This is implied by our loss assumption that

$\text{loss}(y, y) < \text{loss}(y, y')$ for all $y' \neq y$, and hence $\text{argmax}_y f^*_{y^\diamond}(\mathbf{x}, y) = \text{argmin}_y \mathbf{L}_{(y^\diamond, y)}$. So (a) is proved.

We next prove (b). If we assume $\text{argmin}_{y'} \sum_{y \in \mathcal{Y}^\diamond} \alpha_y \text{loss}(y, y') \subseteq \mathcal{Y}^\diamond$ for all $\alpha_{(\cdot)} \geq 0$ and $\sum_{y \in \mathcal{Y}^\diamond} \alpha_y = 1$, then it follows trivially that $\text{argmax}_y f^*(\mathbf{x}, y) \subseteq \mathcal{Y}^\diamond$ for all $f^* \in \mathcal{F}^*_{\mathbf{x}}$. ■

### 5.2 Consistency for Prediction Based on the Predictor Player's Probability

For a prediction task where the set of options a predictor can choose is different from the set of ground truth labels (e.g., the classification task with abstention task in Section 4.3), the analysis in the previous subsection cannot be applied. In this subsection we will establish consistency properties of the adversarial prediction framework for a general loss matrix where the prediction is based on the predictor player's optimal probability.

**Theorem 17** *Given the true distribution $P(Y|\mathbf{x})$ and a loss matrix $\mathbf{L}$, finding the predictor's optimal probability in the adversarial prediction framework reduce to finding the Bayes optimal prediction, assuming that $f$ is allowed to be optimized over all measurable function.*

**Proof** Since the predictor can choose from $l$ options which could be different than the $k$ number of classes in the ground truth, $\mathbf{d}$ and $\mathbf{q}$ lie in the $k$ dimensional probability simplex $\Delta^k$, while the predictor's probability mass $\mathbf{p}$ lies in the $l$ dimensional probability simplex $\Delta^l$. Let $\mathbf{f} \in \mathbb{R}^k$ the vector encoding of the value of $f$ at all classes. The potential function minimizer $f^*$ can now be written as:

$$\mathbf{f}^* \in \operatorname*{argmin}_{\mathbf{f}} \max_{\mathbf{q} \in \Delta^k} \min_{\mathbf{p} \in \Delta^l} \left\{ \mathbf{f}^\intercal \mathbf{q} + \mathbf{p}^\intercal \mathbf{L} \mathbf{q} - \mathbf{d}^\intercal \mathbf{f} \right\}. \tag{18}$$

As noted in our previous analysis, since $\mathbf{f}^*$ is an optimal solution, the objective function must have a zero subgradient at $\mathbf{f}^*$. That means $\mathbf{0} = \mathbf{q}^* - \mathbf{d}$, where $\mathbf{q}^*$ is an optimal solution in Eq. (18) under $\mathbf{f}^*$.

Here we use the probabilistic prediction scheme as mentioned in Eq. (9). The consistency condition in Eq. (11) requires that the loss of this prediction scheme under the optimal potential $\mathbf{f}^*$ and the true probability $\mathbf{d}$ reaches the Bayes optimal risk, i.e.,

$$\mathbf{p}^{\diamond\intercal} \mathbf{L} \mathbf{d} = \min_y (\mathbf{L}\mathbf{d})_y, \quad \text{where} \quad \mathbf{p}^\diamond = \operatorname*{argmin}_{\mathbf{p} \in \Delta^l} \max_{\mathbf{q} \in \Delta^k} \mathbf{p}^\intercal \mathbf{L} \mathbf{q} + \mathbf{f}^{*\intercal} \mathbf{q}.$$

Since the maximization over $\mathbf{q}$ in Eq. (18) does not depend on $\mathbf{d}^\intercal \mathbf{f}$, we know that $\mathbf{d}$ is also an optimal solution of $\text{argmax}_{\mathbf{q} \in \Delta^k} \min_{\mathbf{p} \in \Delta^l} \mathbf{p}^\intercal \mathbf{L} \mathbf{q} + \mathbf{f}^{*\intercal} \mathbf{q}$. Then, based on the minimax duality theorem (Von Neumann and Morgenstern, 1945), we know that:

$$\mathbf{p}^{\diamond\intercal} \mathbf{L} \mathbf{d} + \mathbf{f}^{*\intercal} \mathbf{d} = \min_{\mathbf{p} \in \Delta^l} \mathbf{p}^\intercal \mathbf{L} \mathbf{d} + \mathbf{f}^{*\intercal} \mathbf{d}.$$

This implies that: $\mathbf{p}^{\diamond\intercal} \mathbf{L} \mathbf{d} = \min_{\mathbf{p} \in \Delta^l} \mathbf{p}^\intercal \mathbf{L} \mathbf{d} = \min_y (\mathbf{L}\mathbf{d})_y$, which concludes our proof. ■

## 6. Optimization

The goal of a learning algorithm in the adversarial prediction framework is to obtain the optimal Lagrange dual variable $\theta$ that enforces the adversary's probability distribution to reside within the moment matching constraints in Eq (1). In the risk minimization perspective (Eq. (2)), it is equivalent to finding the parameter $\theta$ that minimizes the adversarial surrogate loss (AL) in Eq. (3). To find the optimal $\theta$, we employ (sub)-gradient methods to optimize our convex objective.

### 6.1 Subgradient-Based Convex Optimization

The risk minimization perspective of adversarial prediction framework (Eq. (2)) can be written as:

$$\min_{\theta} \mathbb{E}_{\mathbf{X},Y\sim\tilde{P}} \left[AL(\mathbf{X},Y,\theta)\right]$$
$$\text{where: } AL(\mathbf{x},y,\theta) = \max_{\mathbf{q}\in\Delta} \min_{\mathbf{p}\in\Delta} \mathbf{p}^\intercal \mathbf{L}\mathbf{q} + \theta^\intercal \left[\sum_j q_j \phi(\mathbf{x},j) - \phi(\mathbf{x},y)\right].$$

The subdifferential of the expected adversarial loss in the objective above is equal to the expected subdifferential of the loss for each sample (Corollary 23.8, Rockafellar, 1970):

$$\partial_\theta \, \mathbb{E}_{\mathbf{X},Y\sim\tilde{P}} \left[AL(\mathbf{X},Y,\theta)\right] = \mathbb{E}_{\mathbf{X},Y\sim\tilde{P}} \left[\partial_\theta \, AL(\mathbf{X},Y,\theta)\right].$$

Theorem 18 describes the subgradient of the adversarial surrogate loss with respect to $\theta$.

**Theorem 18** *Given $\theta$, suppose the set of optimal $\mathbf{q}$ for the maximin inside the AL is $Q^*$:*

$$Q^* = \operatorname*{argmax}_{\mathbf{q}\in\Delta} \min_{\mathbf{p}\in\Delta} \left\{\mathbf{p}^\intercal \mathbf{L}\mathbf{q} + \theta^\intercal \left[\sum_j q_j \phi(\mathbf{x},j) - \phi(\mathbf{x},y)\right]\right\}.$$

*Then the subdifferential of the adversarial loss $AL(\mathbf{x},y,\theta)$ with respect to the parameter $\theta$ can be fully characterized by*

$$\partial_\theta \, AL(\mathbf{x},y,\theta) = \mathbf{conv}\left\{\sum_j q_j^* \phi(\mathbf{x},j) - \phi(\mathbf{x},y) \;\middle|\; \mathbf{q}\in Q^*\right\}.$$

**Proof** Denote $\varphi(\theta,\mathbf{q}) \triangleq \min_{\mathbf{p}\in\Delta} \left\{\mathbf{p}^\intercal \mathbf{L}\mathbf{q} + \theta^\intercal \left[\sum_j q_j \phi(\mathbf{x},j) - \phi(\mathbf{x},y)\right]\right\}$. Then for any fixed $\mathbf{q}$, $\varphi(\theta,\mathbf{q})$ is a closed proper convex function in $\theta$. Denote $g(\theta) \triangleq \max_{\mathbf{q}\in\Delta} \varphi(\theta,\mathbf{q})$. Then the interior of its domain $\operatorname{int}(\operatorname{dom} g)$ is the entire Euclidean space of $\theta$, and $\varphi$ is continuous on $\operatorname{int}(\operatorname{dom} g) \times \Delta$. Using the obvious fact that $\partial_\theta \varphi(\theta,\mathbf{q}) = \left\{\sum_j q_j \phi(\mathbf{x},j) - \phi(\mathbf{x},y)\right\}$, the desired conclusion follows directly from Proposition A.22 of Bertsekas (1971). ■

The runtime complexity to calculate the subgradient of AL for one example above is $\mathcal{O}(k^{3.5})$ due to the need to solve the inner minimiax using linear program (Karmarkar's algorithm). For the loss metrics that we have studied in Section 3 we construct faster ways to compute the subgradient as follows.

**Corollary 19** *The subdifferential of $AL^{0\text{-}1}(\mathbf{x}, y, \theta)$ with respect to $\theta$ includes:*

$$\partial_\theta \; AL^{0\text{-}1}(\mathbf{x}, y, \theta) \ni \tfrac{1}{|S^*|} \textstyle\sum_{j \in S^*} \phi(\mathbf{x}, j) - \phi(\mathbf{x}, y),$$

*where $S^*$ is an optimal solution set of the maximization inside the $AL^{0\text{-}1}$, i.e.:*

$$S^* \in \operatorname*{argmax}_{S \subseteq [k],\, S \neq \emptyset} \frac{\sum_{j \in S} \theta^\intercal \phi(\mathbf{x}, j) + |S| - 1}{|S|}.$$

**Corollary 20** *The subdifferential of $AL^{ord}(\mathbf{x}, y, \theta)$ with respect to $\theta$ includes:*

$$\partial_\theta \; AL^{ord}(\mathbf{x}, y, \theta) \ni \tfrac{1}{2} \left(\phi(\mathbf{x}, i^*) + \phi(\mathbf{x}, j^*)\right) - \phi(\mathbf{x}, y),$$

*where $i^*, j^*$ is the solution of:*

$$(i^*, j^*) \in \operatorname*{argmax}_{i,j \in [k]} \frac{\theta^\intercal \phi(\mathbf{x}, i) + \theta^\intercal \phi(\mathbf{x}, j) + j - i}{2}.$$

**Corollary 21** *The subdifferential of $AL^{abstain}(\mathbf{x}, y, \theta, \alpha)$ where $0 \leq \alpha \leq \frac{1}{2}$ with respect to $\theta$ includes:*

$$\partial_\theta \; AL^{abstain}(\mathbf{x}, y, \theta, \alpha) \ni \begin{cases} (1-\alpha)\phi(\mathbf{x}, i^*) + \alpha\phi(\mathbf{x}, j^*) - \phi(\mathbf{x}, y) & g(\mathbf{x}, y, \theta, \alpha) > h(\mathbf{x}, y, \theta, \alpha) \\ \phi(\mathbf{x}, l^*) - \phi(\mathbf{x}, y) & \textit{otherwise}, \end{cases}$$

*where:*

$$g(\mathbf{x}, y, \theta, \alpha) = \max_{i,j \in [k], i \neq j} (1-\alpha)\, f_i + \alpha f_j + \alpha, \quad h(\mathbf{x}, y, \theta, \alpha) = \max_l f_l,$$
$$(i^*, j^*) \in \operatorname*{argmax}_{i,j \in [k], i \neq j} (1-\alpha)\, f_i + \alpha f_j + \alpha, \quad l^* = \operatorname*{argmax}_l f_l,$$

*and the potential $f_i$ is defined as $f_i = \theta^\intercal \phi(\mathbf{x}, i)$.*

The runtime of the subgradient computation algorithms above are the same as the runtime of computing the adversarial surrogate losses, i.e., $\mathcal{O}(k \log k)$ for $\text{AL}^{\text{0-1}}$, $\mathcal{O}(k)$ for $\text{AL}^{\text{ord}}$, and $\mathcal{O}(k)$ for $\text{AL}^{\text{abstain}}$. This is a significant speed-up compared to the technique that uses a linear program solver.

Since we already have algorithms for computing the subgradient of AL, any subgradient based optimization techniques can be used to optimize $\theta$ including some stochastic (sub)-gradient techniques like SGD, AdaGrad, and ADAM or batch (sub)-gradient techniques like L-BFGS. Some regularization techniques such as L1 and L2 regularizations, can also be added to the objective function. The optimization is guaranteed to converge to the global optimum as the objective is convex.

### 6.2 Incorporating Rich Feature Spaces via the Kernel Trick

Considering large feature spaces is important for developing an expressive classifier that can learn from large amounts of training data. Indeed, Fisher consistency requires such feature spaces for its guarantees to be meaningful. However, naïvely projecting from the original feature space, $\phi(\mathbf{x}, y)$, to a richer (or possibly infinite) feature space $\omega(\phi(\mathbf{x}, y))$, can be computationally burdensome. Kernel methods enable this feature expansion by allowing the dot products of certain feature functions to be computed implicitly, i.e., $K(\phi(\mathbf{x}_i, y_i), \phi(\mathbf{x}_j, y_j)) = \omega(\phi(\mathbf{x}_i, y_i)) \cdot \omega(\phi(\mathbf{x}_j, y_j))$.

To formulate a learning algorithm for adversarial surrogate losses that can incorporate richer feature spaces via kernel trick, we apply the PEGASOS algorithm (Shalev-Shwartz et al., 2011) to our losses. Instead of optimizing the problem in the dual formulation as in many kernel trick algorithms, PEGASOS allows us to incorporate the kernel trick into its primal stochastic subgradient optimization technique. The algorithm works on L2 penalized risk minimization,

$$\min_{\theta} \mathbb{E}_{\mathbf{X},Y\sim\tilde{P}} \frac{\lambda}{2}\|\theta\|^2 + AL(\mathbf{X}, Y, \theta),$$

where $\lambda$ is the regularization penalty parameter. Since we want to perform stochastic optimization, we replace the objective above with an approximation based on a single training example:

$$\frac{\lambda}{2}\|\theta\|^2 + AL(\mathbf{x}_{i_t}, y_{i_t}, \theta),$$

where $i_t$ indicates the index of the example randomly selected at iteration $t$. Therefore at iteration $t$, the subgradient of our objective function with respect to the parameter $\theta$ is:

$$\begin{aligned} \partial_\theta^{(t)} =& \lambda\theta^{(t)} + \textstyle\sum_j q_j^{*(t)}\phi(\mathbf{x}_{i_t}, j) - \phi(\mathbf{x}_{i_t}, y_{i_t}), \\ \text{where: } \mathbf{q}^{*(t)} =& \operatorname*{argmax}_{\mathbf{q}\in\Delta} \min_{\mathbf{p}\in\Delta} \mathbf{p}^\intercal\mathbf{L}\mathbf{q} + \mathbf{f}^{(t)\intercal}\mathbf{q} - f_{y_{i_t}}^{(t)}, \\ f_j^{(t)} =& \theta^{(t)\intercal}\phi(\mathbf{x}_{i_t}, j). \end{aligned} \tag{19}$$

The algorithm starts with zero initialization, i.e., $\theta^{(1)} = \mathbf{0}$ and uses a pre-determined learning rate scheme $\eta^{(t)} = \frac{1}{\lambda t}$ to take optimization steps,

$$\theta^{(t+1)} = \theta^{(t)} - \eta^{(t)}\partial_\theta^{(t)} = \theta^{(t)} - \tfrac{1}{\lambda t}\partial_\theta^{(t)}.$$

Let us denote $\mathbf{g}^{(t)} = \sum_j q_j^{*(t)}\phi(\mathbf{x}_{i_t}, j) - \phi(\mathbf{x}_{i_t}, y_{i_t})$ from Eq. (19), then the update steps can be written as:

$$\theta^{(t+1)} = (1 - \tfrac{1}{t})\theta^{(t)} - \tfrac{1}{\lambda t}\mathbf{g}^{(t)}.$$

By accumulating the weighted contribution of $\mathbf{g}$ for each step, the value of $\theta$ at iteration $t+1$ is:

$$\theta^{(t+1)} = -\frac{1}{\lambda t}\sum_{l=1}^{t}\mathbf{g}^{(l)},$$

which can be expanded to the original formulation of our subgradient:

$$\theta^{(t+1)} = -\frac{1}{\lambda t}\sum_{l=1}^{t}\sum_{j=1}^{k} q_j^{*(l)}\phi(\mathbf{x}_{i_l}, j) - \phi(\mathbf{x}_{i_l}, y_{i_l}), \tag{20}$$

$$\text{where} \quad \mathbf{q}^{*(l)} = \operatorname*{argmax}_{\mathbf{q}\in\Delta} \min_{\mathbf{p}\in\Delta} \mathbf{p}^\intercal \mathbf{L}\mathbf{q} + \mathbf{f}^{(l)\intercal}\mathbf{q} - f^{(l)}_{y_{i_l}},$$

$$f_j^{(l)} = \theta^{(l)\intercal}\phi(\mathbf{x}_{i_l}, j).$$

Let $\mathbf{z}$ be the one-hot vector representation of the ground truth label $y$ where its elements are $z_y = 1$, and $z_j = 0$ for all $j \neq y$. From the definition of $\mathbf{g}^{(t)}$, let us denote $\mathbf{r}^{(t)} = \mathbf{q}^{*(t)} - \mathbf{z}_{i_t}$, then $\mathbf{g}^{(t)}$ can be equivalently written as $\mathbf{g}^{(t)} = \sum_j {r_j}^{(t)}\phi(\mathbf{x}_{i_t}, j)$. We denote $\boldsymbol{\alpha}_i^{(t+1)}$ as a vector that accumulates the value of $\mathbf{r}$ for the $i$-th example each time it is selected until iteration $t$. Then, the value of $\theta^{(t+1)}$ in Eq. (20) can be equivalently written as:

$$\theta^{(t+1)} = -\frac{1}{\lambda t}\sum_{i=1}^{n}\sum_{j=1}^{k}\alpha_{(i,j)}^{(t+1)}\phi(\mathbf{x}_i, j),$$

where $\alpha_{(i,j)}^{(t+1)}$ indicates the $j$-th element of the vector $\boldsymbol{\alpha}_i^{(t+1)}$. Using this notation, the potentials $\mathbf{f}^{(t)}$ used to calculate the adversarial loss can be computed as:

$$f_j^{(t)} = \theta^{(t)\intercal}\phi(\mathbf{x}_{i_t}, j) = -\frac{1}{\lambda t}\sum_{i'}^{n}\sum_{j'}^{k}\alpha_{(i',j')}^{(t)}\,\phi(\mathbf{x}_{i'}, j')\cdot\phi(\mathbf{x}_{i_t}, j).$$

Note that the computation of the potentials above only depends on the dot product between the feature functions weighted by the $\boldsymbol{\alpha}$ variables.

Since the algorithm only depends on the dot products, to incorporate a richer feature spaces $\omega(\phi(\mathbf{x}, y))$, we can directly apply kernel function in the computation of the potentials,

$$\begin{aligned} f_j^{(t)} = \theta^{(t)\intercal}\omega(\phi(\mathbf{x}_{i_t}, j)) &= -\frac{1}{\lambda t}\sum_{i'}^{n}\sum_{j'}^{k}\alpha_{(i',j')}^{(t)}\omega(\phi(\mathbf{x}_{i'}, j'))\cdot\omega(\phi(\mathbf{x}_{i_t}, j)) \\ &= -\frac{1}{\lambda t}\sum_{i'}^{n}\sum_{j'}^{k}\alpha_{(i',j')}^{(t)}K(\phi(\mathbf{x}_{i'}, j'), \phi(\mathbf{x}_{i_t}, j)). \end{aligned}$$

The detailed algorithm for our adversarial surrogate loss is described in Algorithm 1.

## 7. Experiments

We conduct experiments on real data to investigate the empirical performance of the adversarial surrogate losses in several prediction tasks.

### 7.1 Experiments for Multiclass Zero-One Loss Metric

We evaluate the performance of the $\text{AL}^{\text{0-1}}$ classifier and compare it with the three most popular multiclass SVM formulations: WW (Weston et al., 1999), CS (Crammer and Singer,

**Algorithm 1** PEGASOS algorithm for adversarial surrogate losses with kernel trick

**Input:** Training data $(\mathbf{x}_1, y_1), \dots (\mathbf{x}_n, y_n)$, $\mathbf{L}, \lambda, T, k$
$\boldsymbol{\alpha}_i^{(1)} \leftarrow \mathbf{0}, \forall i \in \{1, \dots, n\}$
Let $\mathbf{z}_i$ be the one-hot encoding of $y_i$ for all $i \in \{1, \dots, n\}$
**for** $t \leftarrow 1, 2, \dots, T$ **do**
  Choose $i_t \in \{1, \dots, n\}$ uniformly at random
  Compute $\mathbf{f}^{(t)}$, where $f_j^{(t)} \leftarrow -\frac{1}{\lambda t} \sum_{i'}^{n} \sum_{j'}^{k} \alpha_{(i',j')}^{(t)} K(\phi(\mathbf{x}_{i'}, j'), \phi(\mathbf{x}_{i_t}, j))$
  $\mathbf{q}^{*(t)} \leftarrow \operatorname{argmax}_{\mathbf{q} \in \Delta} \min_{\mathbf{p} \in \Delta} \mathbf{p}^\intercal \mathbf{L} \mathbf{q} + \mathbf{f}^{(t)\intercal} \mathbf{q} - f_{y_{i_t}}^{(t)}$
  $\boldsymbol{\alpha}_{i_t}^{(t+1)} \leftarrow \boldsymbol{\alpha}_{i_t}^{(t)} + \mathbf{q}^{*(t)} - \mathbf{z}_{i_t}$
**end for**
**return** $\boldsymbol{\alpha}_i^{(t+1)}, \ \forall i \in \{1, \dots, n\}$

2002), and LLW (Lee et al., 2004). We use 12 datasets from the UCI machine learning repository (Lichman, 2013) with various sizes and numbers of classes (details in Table 1). For each dataset, we consider the methods using the original feature space (linear kernel) and a kernelized feature space using the Gaussian radial basis function kernel.

| Dataset | Properties | | | |
|---|---|---|---|---|
| | #class | #train | # test | #feature |
| (1) iris | 3 | 105 | 45 | 4 |
| (2) glass | 6 | 149 | 65 | 9 |
| (3) redwine | 10 | 1119 | 480 | 11 |
| (4) ecoli | 8 | 235 | 101 | 7 |
| (5) vehicle | 4 | 592 | 254 | 18 |
| (6) segment | 7 | 1617 | 693 | 19 |
| (7) sat | 7 | 4435 | 2000 | 36 |
| (8) optdigits | 10 | 3823 | 1797 | 64 |
| (9) pageblocks | 5 | 3831 | 1642 | 10 |
| (10) libras | 15 | 252 | 108 | 90 |
| (11) vertebral | 3 | 217 | 93 | 6 |
| (12) breasttissue | 6 | 74 | 32 | 9 |

Table 1: Properties of the datasets for the zero-one loss metric experiments.

For our experimental methodology, we first make 20 random splits of each dataset into training and testing sets. We then perform two-stage, five-fold cross validation on the training set of the first split to tune each model's parameter $C$ and the kernel parameter $\gamma$ under the kernelized formulation. In the first stage, the values for $C$ are $2^i, i = \{0, 3, 6, 9, 12\}$ and the values for $\gamma$ are $2^i, i = \{-12, -9, -6, -3, 0\}$. We select final values for $C$ from $2^i C_0, i = \{-2, -1, 0, 1, 2\}$ and values for $\gamma$ from $2^i \gamma_0, i = \{-2, -1, 0, 1, 2\}$ in the second stage, where $C_0$ and $\gamma_0$ are the best parameters obtained in the first stage. Using the selected parameters, we train each model on the 20 training sets and evaluate the performance on the corresponding testing set. We use the Shark machine learning library (Igel et al., 2008) for the implementation of the three multiclass SVM formulations.

| D | Linear Kernel | | | | Gaussian Kernel | | | |
|---|---|---|---|---|---|---|---|---|
| | $AL^{0-1}$ | WW | CS | LLW | $AL^{0-1}$ | WW | CS | LLW |
| (1) | **96.3** (3.1) | **96.0** (2.6) | **96.3** (2.4) | 79.7 (5.5) | **96.7** (2.4) | **96.4** (2.4) | **96.2** (2.3) | 95.4 (2.1) |
| (2) | **62.5** (6.0) | **62.2** (3.6) | **62.5** (3.9) | 52.8 (4.6) | **69.5** (4.2) | 66.8 (4.3) | **69.4** (4.8) | **69.2** (4.4) |
| (3) | **58.8** (2.0) | **59.1** (1.9) | 56.6 (2.0) | 57.7 (1.7) | 63.3 (1.8) | 64.2 (2.0) | 64.2 (1.9) | **64.7** (2.1) |
| (4) | **86.2** (2.2) | 85.7 (2.5) | **85.8** (2.3) | 74.1 (3.3) | **86.0** (2.7) | 84.9 (2.4) | **85.6** (2.4) | **86.0** (2.5) |
| (5) | **78.8** (2.2) | **78.8** (1.7) | **78.4** (2.3) | 69.8 (3.7) | **84.3** (2.5) | **84.4** (2.6) | 83.8 (2.3) | **84.4** (2.6) |
| (6) | 94.9 (0.7) | 94.9 (0.8) | **95.2** (0.8) | 75.8 (1.5) | **96.5** (0.6) | **96.6** (0.5) | 96.3 (0.6) | **96.4** (0.5) |
| (7) | 84.9 (0.7) | **85.4** (0.7) | 84.7 (0.7) | 74.9 (0.9) | **91.9** (0.5) | **92.0** (0.6) | **91.9** (0.5) | **91.9** (0.4) |
| (8) | **96.6** (0.6) | **96.5** (0.7) | 96.3 (0.6) | 76.2 (2.2) | 98.7 (0.4) | **98.8** (0.4) | **98.8** (0.3) | **98.9** (0.3) |
| (9) | 96.0 (0.5) | 96.1 (0.5) | **96.3** (0.5) | 92.5 (0.8) | **96.8** (0.5) | 96.6 (0.4) | 96.7 (0.4) | 96.6 (0.4) |
| (10) | **74.1** (3.3) | 72.0 (3.8) | 71.3 (4.3) | 34.0 (6.4) | 83.6 (3.8) | 83.8 (3.4) | **85.0** (3.9) | 83.2 (4.2) |
| (11) | **85.5** (2.9) | **85.9** (2.7) | **85.4** (3.3) | 79.8 (5.6) | **86.0** (3.1) | **85.3** (2.9) | 85.5 (3.3) | 84.4 (2.7) |
| (12) | **64.4** (7.1) | 59.7 (7.8) | **66.3** (6.9) | 58.3 (8.1) | **68.4** (8.6) | **68.1** (6.5) | **66.6** (8.9) | **68.0** (7.2) |
| avg | 81.59 | 81.02 | 81.25 | 68.80 | 85.14 | 84.82 | 85.00 | 84.93 |
| #b | 9 | 7 | 8 | 0 | 9 | 7 | 7 | 8 |

Table 2: The mean and (in parentheses) standard deviation of the accuracy for each model with linear kernel and Gaussian kernel feature representations. Bold numbers in each case indicate that the result is the best or not significantly worse than the best (Wilcoxon signed-rank test with $\alpha = 0.05$).

We report the accuracy of each method averaged over the 20 dataset splits for both linear feature representations and Gaussian kernel feature representations in Table 2. We denote the results that are either the best of all four methods or not worse than the best with statistical significance (under the non-parametric Wilcoxon signed-rank test with $\alpha = 0.05$) using bold font. We also show the accuracy averaged over all of the datasets for each method and the number of dataset for which each method is "indistinguishably best" (bold numbers) in the last row. As we can see from the table, the only alternative model that is Fisher consistent—the LLW model—performs poorly on all datasets when only linear features are employed. This matches with previous experimental results conducted by Doğan et al. (2016) and demonstrates a weakness of using an absolute margin for the loss function (rather than the relative margins of all other methods). The $AL^{0-1}$ classifier performs competitively with the WW and CS models with a slight advantages on overall average accuracy and a larger number of "indistinguishably best" performances on datasets—or, equivalently, fewer statistically significant losses to any other method.

The kernel trick in the Gaussian kernel case provides access to much richer feature spaces, improving the performance of all models, and the LLW model especially. In general, all models provide competitive results in the Gaussian kernel case. The $AL^{0-1}$ classifier maintains a similarly slight advantage and only provides performance that is sub-optimal (with statistical significance) in three of the twelve datasets versus six of twelve and five of twelve for the other methods. We conclude that the multiclass adversarial method performs well in both low and high dimensional feature spaces. Recalling the theoretical analysis of the adversarial method, it is a well-motivated (from the adversarial zero-one loss minimiza-

tion) multiclass classifier that enjoys both strong theoretical properties (Fisher consistency) and empirical performance.

### 7.2 Experiments for Multiclass Ordinal Classification

We conduct our ordinal classification experiments on a benchmark dataset for ordinal regression (Chu and Ghahramani, 2005), evaluate the performance using mean absolute error (MAE), and perform statistical tests on the results of different hinge loss surrogate methods. The benchmark contains datasets taken from the UCI machine learning repository (Lichman, 2013), which range from relatively small to relatively large datasets. The characteristic of the datasets, i.e., the number of classes, the training set size, the testing set size, and the number of features is described in Table 3.

| Dataset | #class | #train | #test | #features |
|---|---|---|---|---|
| diabetes | 5 | 30 | 13 | 2 |
| pyrimidines | 5 | 51 | 23 | 27 |
| triazines | 5 | 130 | 56 | 60 |
| wisconsin | 5 | 135 | 59 | 32 |
| machinecpu | 10 | 146 | 63 | 6 |
| autompg | 10 | 274 | 118 | 7 |
| boston | 5 | 354 | 152 | 13 |
| stocks | 5 | 665 | 285 | 9 |
| abalone | 10 | 2923 | 1254 | 10 |
| bank | 10 | 5734 | 2458 | 8 |
| computer | 10 | 5734 | 2458 | 21 |
| calhousing | 10 | 14447 | 6193 | 8 |

Table 3: Properties of the datasets for the ordinal classification experiments.

In the experiment, we consider the methods using the original feature space and using a Gaussian radial basis function kernel feature space. The methods that we compare include two variations of our approach, the threshold based ($\text{AL}^{\text{ord-th}}$), and the multiclass-based ($\text{AL}^{\text{ord-mc}}$). The baselines we use for the threshold-based models include an SVM-based reduction framework algorithm ($\text{RED}^{\text{th}}$) (Li and Lin, 2007), the *all threshold* method with hinge loss (AT) (Shashua and Levin, 2003; Chu and Keerthi, 2005), and the *immediate threshold* method with hinge loss (IT) (Shashua and Levin, 2003; Chu and Keerthi, 2005). For the multiclass-based models, we compare our method with an SVM-based reduction framework algorithm using multiclass features ($\text{RED}^{\text{mc}}$) (Li and Lin, 2007), cost-sensitive one-sided support vector regression (CSOSR) (Tu and Lin, 2010), cost-sensitive one-versus-one SVM (CSOVO) (Lin, 2014), and cost-sensitive one-versus-all SVM (CSOVA) (Lin, 2008). For our Gaussian kernel experiment, we compare our threshold-based model ($\text{AL}^{\text{ord-th}}$) with SVORIM and SVOREX (Chu and Keerthi, 2005).

In our experiments, we first make 20 random splits of each dataset into training and testing sets. We performed two stages of five-fold cross validation on the first split training set for tuning each model's regularization constant $\lambda$. In the first stage, the possible values for $\lambda$ are $2^{-i}, i = \{1, 3, 5, 7, 9, 11, 13\}$. Using the best $\lambda$ in the first stage, we set the possible values for $\lambda$ in the second stage as $2^{\frac{i}{2}}\lambda_0, i = \{-3, -2, -1, 0, 1, 2, 3\}$, where $\lambda_0$ is the

best parameter obtained in the first stage. Using the selected parameter from the second stage, we train each model on the 20 training sets and evaluate the MAE performance on the corresponding testing set. We then perform a statistical test to find whether the performance of a model is different with statistical significance from other models. Similarly, we perform the Gaussian kernel experiments with the same model parameter settings as in the multiclass zero-one experiments.

We report the mean absolute error (MAE) averaged over the dataset splits as shown in Table 4 and Table 5. We highlight the results that are either the best or not worse than the best with statistical significance (under the non-parametric Wilcoxon signed-rank test with $\alpha = 0.05$) in boldface font. We also provide the summary for each model in terms of the averaged MAE over all datasets and the number of datasets for which each model marked with boldface font in the bottom of the table.

| Dataset | Threshold-based models | | | | Multiclass-based models | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $AL^{ord\text{-}th}$ | $RED^{th}$ | AT | IT | $AL^{ord\text{-}mc}$ | $RED^{mc}$ | CSOSR | CSOVO | CSOVA |
| diabetes | **0.696** | **0.715** | 0.731 | 0.827 | **0.692** | **0.700** | **0.715** | 0.738 | **0.762** |
| | (0.13) | (0.19) | (0.15) | (0.28) | (0.14) | (0.15) | (0.19) | (0.16) | (0.19) |
| pyrimidines | 0.654 | 0.678 | 0.615 | 0.626 | **0.509** | 0.565 | **0.520** | 0.576 | **0.526** |
| | (0.12) | (0.15) | (0.3) | (0.14) | (0.12) | (0.13) | (0.13) | (0.16) | (0.16) |
| triazines | **0.607** | 0.683 | 0.649 | 0.654 | 0.670 | 0.673 | 0.677 | 0.738 | 0.732 |
| | (0.09) | (0.11) | (0.11) | (0.12) | (0.09) | (0.11) | (0.10) | (0.10) | (0.10) |
| wisconsin | **1.077** | **1.067** | **1.097** | 1.175 | 1.136 | 1.141 | 1.208 | 1.275 | 1.338 |
| | (0.11) | (0.12) | (0.11) | (0.14) | (0.11) | (0.10) | (0.12) | (0.15) | (0.11) |
| machinecpu | **0.449** | **0.456** | **0.458** | **0.467** | 0.518 | 0.515 | 0.646 | 0.602 | 0.702 |
| | (0.09) | (0.09) | (0.09) | (0.10) | (0.11) | (0.10) | (0.10) | (0.09) | (0.14) |
| autompg | **0.551** | **0.550** | **0.550** | 0.617 | 0.599 | 0.602 | 0.741 | 0.598 | 0.731 |
| | (0.06) | (0.06) | (0.06) | (0.07) | (0.06) | (0.06) | (0.07) | (0.06) | (0.07) |
| boston | 0.316 | **0.304** | **0.306** | **0.298** | 0.311 | 0.311 | 0.353 | **0.294** | 0.363 |
| | (0.03) | (0.03) | (0.03) | (0.04) | (0.03) | (0.04) | (0.05) | (0.04) | (0.04) |
| stocks | 0.324 | 0.317 | 0.315 | 0.324 | 0.168 | 0.175 | 0.204 | **0.147** | 0.213 |
| | (0.02) | (0.02) | (0.02) | (0.02) | (0.02) | (0.03) | (0.02) | (0.02) | (0.02) |
| abalone | 0.551 | 0.547 | 0.546 | 0.571 | **0.521** | **0.520** | 0.545 | 0.558 | 0.556 |
| | (0.02) | (0.02) | (0.02) | (0.02) | (0.02) | (0.02) | (0.02) | (0.02) | (0.02) |
| bank | 0.461 | 0.460 | 0.461 | 0.461 | **0.445** | **0.446** | 0.732 | 0.448 | 0.989 |
| | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) | (0.02) | (0.01) | (0.02) |
| computer | 0.640 | 0.635 | 0.633 | 0.683 | **0.625** | **0.624** | 0.889 | 0.649 | 1.055 |
| | (0.02) | (0.02) | (0.02) | (0.02) | (0.01) | (0.02) | (0.02) | (0.02) | (0.02) |
| calhousing | 1.190 | 1.183 | 1.182 | 1.225 | 1.164 | **1.144** | 1.237 | 1.202 | 1.601 |
| | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) | (0.01) | (0.02) |
| average | 0.626 | 0.633 | 0.629 | 0.661 | 0.613 | 0.618 | 0.706 | 0.652 | 0.797 |
| # bold | 5 | 5 | 4 | 2 | 5 | 5 | 2 | 2 | 2 |

Table 4: The average and (in parenthesis) standard deviation of the mean absolute error (MAE) for each model. Bold numbers in each case indicate that the result is the best or not significantly worse than the best (Wilcoxon signed-rank test with $\alpha = 0.05$).

As we can see from Table 4, in the experiment with the original feature space, threshold-based models perform well on relatively small datasets, whereas multiclass-based models perform well on relatively large datasets. A possible explanation for this result is that multiclass-based models have more flexibility in creating decision boundaries, hence perform better if the training data size is sufficient. However, since multiclass-based models have many more parameters than threshold-based models ($mk$ parameters rather than $m+k-1$ parameters), multiclass methods may need more data, and hence, may not perform well on relatively small datasets.

In the threshold-based models comparison, $\text{AL}^{\text{ord-th}}$, $\text{RED}^{\text{th}}$, and AT perform competitively on relatively small datasets like `triazines`, `wisconsin`, `machinecpu`, and `autompg`. $\text{AL}^{\text{ord-th}}$ has a slight advantage over $\text{RED}^{\text{th}}$ on the overall accuracy, and a slight advantage over AT on the number of "indistinguishably best" performance on all datasets. We can also see that AT is superior to IT in the experiments under the original feature space. Among the multiclass-based models, $\text{AL}^{\text{ord-mc}}$ and $\text{RED}^{\text{mc}}$ perform competitively on datasets like `abalone`, `bank`, and `computer`, with a slight advantage of $\text{AL}^{\text{ord-mc}}$ model on the overall accuracy. In general, the cost-sensitive models perform poorly compared with $\text{AL}^{\text{ord-mc}}$ and $\text{RED}^{\text{mc}}$. A notable exception is the CSOVO model which perform very well on the `stocks`, and `boston` datasets.

| Dataset | $\text{AL}^{\text{ord-th}}$ | SVORIM | SVOREX |
|---|---|---|---|
| diabetes | **0.696** (0.13) | **0.665** (0.14) | **0.688** (0.18) |
| pyrimidines | **0.478** (0.11) | 0.539 (0.11) | 0.550 (0.11) |
| triazines | **0.608** (0.08) | **0.612** (0.09) | **0.604** (0.08) |
| wisconsin | **1.090** (0.10) | 1.113 (0.12) | **1.049** (0.09) |
| machinecpu | **0.452** (0.09) | 0.652 (0.12) | 0.628 (0.13) |
| autompg | **0.529** (0.04) | 0.589 (0.05) | 0.593 (0.05) |
| boston | **0.278** (0.04) | 0.324 (0.03) | 0.316 (0.03) |
| stocks | **0.103** (0.02) | **0.099** (0.01) | **0.100** (0.02) |
| average | 0.531 | 0.574 | 0.566 |
| # bold | 8 | 3 | 4 |

Table 5: The mean and (in parenthesis) standard deviation of the MAE for models with Gaussian kernel. Bold numbers in each case indicate that the result is the best or not significantly worse than the best (Wilcoxon signed-rank test with $\alpha = 0.05$).

In the Gaussian kernel experiment, we can see from Table 5 that the kernelized version of $\text{AL}^{\text{ord-th}}$ performs significantly better than the threshold-based models SVORIM and SVOREX in terms of both the overall accuracy and the number of "indistinguishably best" performance on all datasets. We also note that immediate-threshold-based model (SVOREX) performs better than all-threshold-based model (SVORIM) in our experiment using Gaussian kernel. We can conclude that our proposed adversarial losses for ordinal regression perform competitively compared to the state-of-the-art ordinal regression models using both original feature spaces and kernel feature spaces with a significant performance improvement in the Gaussian kernel experiments.

### 7.3 Experiments for the Classification with Abstention

We conduct experiments for classification with abstention tasks using the same dataset as in the multiclass zero-one experiments (Table 1). We compare the performance of our adversarial surrogate loss ($\text{AL}^{\text{abstain}}$) with the SVM's one-vs-all (OVA) and Crammer & Singer (CS) formulations for classification with abstention (Ramaswamy et al., 2018). We evaluate the prediction performance for a $k$-class classification using the abstention loss:

$$\text{loss}(\hat{y}, y) = \begin{cases} \alpha & \hat{y} = k+1 \\ I(\hat{y} \neq y) & \text{otherwise,} \end{cases}$$

where $\hat{y} = k+1$ indicates an abstain prediction, and $\alpha$ is a fixed value for the penalty for making abstain prediction. Throughout the experiments, we use the standard value of $\alpha = \frac{1}{2}$.

Similar to the setup in the previous experiments, we make 20 random splits of each dataset into training and testing sets. We then perform two-stage, five-fold cross validation on the training set of the first split to tune each model's parameter ($C$ or $\lambda$) and the kernel parameter $\gamma$ under the kernelized formulation. Using the selected parameters, we train each model on the 20 training sets and evaluate the performance on the corresponding testing set. In the prediction step, we use a non-probabilistic prediction scheme for $\text{AL}^{\text{abstain}}$ as presented in Corollary 14. For the baseline methods, we use a threshold base prediction scheme as presented in (Ramaswamy et al., 2018) with the default value of the threshold $\tau$ for each model ($\tau = 0.5$ for the SVM-CS, and $\tau = 0$ for the SVM-OVA).

We report the abstention loss averaged over the dataset splits as shown in Table 6. We highlight the results that are either the best or not worse than the best with statistical significance (under the non-parametric Wilcoxon signed-rank test with $\alpha = 0.05$) in boldface font. We also report the average percentage of abstain predictions produced by each model in each dataset. Finally, we provide the summary for each model in terms of the averaged abstention loss over all datasets and the number of datasets for which each model is marked with boldface font in the bottom of the table.

The results from Table 6 indicates that all models output more abstain predictions in the case of the dataset with higher noise (i.e., bigger value of loss). The percentage of abstain predictions of $\text{AL}^{\text{abstain}}$, SVM-OVA, and SVM-CS are fairly similar. In some datasets like `segment` and `pageblocks`, all models output very rarely abstain, whereas in some datasets like `redwine` and `breasttissue`, some of the models abstain for more than 50% of the total number of testing examples. The results show that this percentage does not depend on the number of classes. For example, both `redwine` and `optdigits` are 10-class classification problems. However, the percentage of abstain prediction for `optdigits` is far less than the one for `redwine`.

In the linear kernel experiments, the $\text{AL}^{\text{abstain}}$ performs best compared the baselines in terms of the overall abstention loss and the number of "indistinguishably best" performance, followed by SVM-CS and then SVM-OVA. The $\text{AL}^{\text{abstain}}$ has a slight advantage compared with the SVM-CS in most of the datasets in the linear kernel experiments except in few datasets that the $\text{AL}^{\text{abstain}}$ outperfoms the SVM-CS by significant margins. Overall, the SVM-OVA performs poorly on most datasets except in a few datasets (`libras`, `vertebral`, and `breasttissue`).

| Dataset | Linear Kernel | | | Gaussian Kernel | | |
|---|---|---|---|---|---|---|
| | $\text{AL}^{\text{abstain}}$ | OVA | CS | $\text{AL}^{\text{abstain}}$ | OVA | CS |
| iris | **0.037** (0.02) [7%] | 0.122 (0.04) [13%] | **0.038** (0.02) [6%] | **0.051** (0.03) [6%] | 0.120 (0.04) [14%] | **0.043** (0.03) [1%] |
| glass | **0.380** (0.04) [40%] | **0.393** (0.04) [27%] | **0.379** (0.04) [38%] | **0.302** (0.03) [37%] | 0.393 (0.04) [35%] | **0.317** (0.03) [25%] |
| redwine | **0.418** (0.01) [58%] | 0.742 (0.04) [50%] | 0.423 (0.01) [54%] | **0.373** (0.01) [42%] | 0.742 (0.04) [50%] | 0.391 (0.01) [58%] |
| ecoli | **0.165** (0.02) [17%] | 0.222 (0.10) [11%] | **0.213** (0.10) [15%] | 0.160 (0.03) [17%] | 0.221 (0.10) [11%] | **0.144** (0.02) [5%] |
| vehicle | **0.214** (0.02) [23%] | 0.231 (0.02) [17%] | **0.216** (0.02) [20%] | **0.206** (0.03) [20%] | 0.226 (0.03) [15%] | 0.300 (0.02) [31%] |
| segment | 0.061 (0.01) [7%] | 0.082 (0.01) [11%] | **0.052** (0.01) [6%] | **0.042** (0.01) [5%] | 0.084 (0.01) [11%] | 0.102 (0.01) [13%] |
| sat | **0.147** (0.01) [14%] | 0.356 (0.01) [20%] | 0.337 (0.01) [14%] | **0.094** (0.01) [9%] | 0.356 (0.01) [20%] | 0.181 (0.01) [4%] |
| optdigits | **0.037** (0.01) [4%] | 0.045 (0.01) [5%] | 0.038 (0.01) 5% | 0.062 (0.01) [12%] | **0.051** (0.01) [5%] | 0.072 (0.01) [8%] |
| pageblocks | **0.040** (0.01) [3%] | 0.042 (0.01) [1%] | **0.045** (0.02) [4%] | **0.037** (0.01) [4%] | 0.042 (0.01) [1%] | 0.060 (0.01) [4%] |
| libras | **0.260** (0.03) [36%] | **0.253** (0.02) [36%] | **0.253** (0.02) [36%] | 0.263 (0.02) [50%] | 0.362 (0.04) [4%] | **0.207** (0.03) [14%] |
| vertebral | 0.154 (0.02) [16%] | **0.147** (0.02) [7%] | 0.159 (0.02) [14%] | 0.181 (0.02) [22%] | **0.147** (0.03) [7%] | 0.220 (0.04) [4%] |
| breasttissue | **0.315** (0.04) [51%] | **0.316** (0.05) [37%] | **0.326** (0.06) [32%] | **0.330** (0.04) [54%] | **0.313** (0.06) [32%] | 0.367 (0.03) [67%] |
| average | 0.186 | 0.246 | 0.207 | 0.175 | 0.255 | 0.200 |
| # bold | 10 | 4 | 8 | 8 | 3 | 4 |

Table 6: The mean and (in parentheses) standard deviation of the abstention loss, and (in square bracket) the percentage of abstain predictions for each model with linear kernel and Gaussian kernel feature representations. Bold numbers in each case indicate that the result is the best or not significantly worse than the best (Wilcoxon signed-rank test with $\alpha = 0.05$).

The introduction of non-linearity via the Gaussian kernel improves the performance of both $\text{AL}^{\text{abstain}}$ and SVM-CS as we see from Table 6. The $\text{AL}^{\text{abstain}}$ method maintains its advantages over the baselines in terms of the overall abstention loss and the number of "indistinguishably best" performances. We can conclude that $\text{AL}^{\text{abstain}}$ performs competitively compared to the baseline models using both original feature spaces and the Gaussian kernel feature spaces. We note that these competitive advantages do not have any drawbacks in terms of the computational cost compared to the baselines. As described in Section 3.5 and Section 4.3, the surrogate loss function and prediction rule are relatively simple and easy to compute.

## 8. Conclusions

In this paper, we proposed an adversarial prediction framework for general multiclass classification that seeks a predictor distribution that robustly optimizes non-convex and non-continuous multiclass loss metrics against the worst-case conditional label distributions (the adversarial distribution) constrained to (approximately) match the statistics of the training data. The dual formulation of the framework resembles a risk minimization model with a convex surrogate loss we call *the adversarial surrogate loss*. These adversarial surrogate losses provide desirable properties of surrogate losses for multiclass classification. For example, in the case of multiclass zero-one classification, our surrogate loss fills the long-standing gap in multiclass classication by simultaneously: guaranteeing Fisher consistency, enabling computational efciency via the kernel trick, and providing competitive performance in practice. Our formulations for the ordinal classification problem provide novel consistent surrogate losses that have not previously been considered in the literature. Lastly, our surrogate loss for the classification with abstention problem provides a unique consistent method that is applicable to binary and multiclass problems, fast to compute, and also competitive in practice.

In general, we showed that the adversarial surrogate losses for general multiclass classification problems enjoy the nice theoretical property of Fisher consistency. We also developed efficient algorithms for optimizing the surrogate losses and a way to incorporate rich feature representation via kernel tricks. Finally, we demonstrated that the adversarial surrogate losses provide competitive performance in practice on several datasets taken from UCI machine learning repository. We will investigate the adversarial prediction framework for more general loss metrics (e.g., multivariate loss metrics), and also for different prediction settings (e.g., active learning and multitask learning) in our future works.

## Acknowledgments

This research was supported as part of the Future of Life Institute (futureoflife.org) FLI-RFP-AI1 program, grant#2016-158710 and by NSF grant RI-#1526379.

## References

Niclas Andréasson, Anton Evgrafov, Michael Patriksson, Emil Gustavsson, and Magnus Önnheim. *An introduction to continuous optimization: foundations and fundamental algorithms*, volume 28. Studentlitteratur Lund, 2005.

Kaiser Asif, Wei Xing, Sima Behpour, and Brian D. Ziebart. Adversarial cost-sensitive classification. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2015.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Evaluation measures for ordinal regression. In *2009 Ninth International Conference on Intelligent Systems Design and Applications*, pages 283–287. IEEE, 2009.

Peter L Bartlett and Marten H Wegkamp. Classification with a reject option using a hinge loss. *The Journal of Machine Learning Research*, 9:1823–1840, 2008.

Peter L. Bartlett, Michael I. Jordan, and Jon D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Dimitri P. Bertsekas. *Control of Uncertain Systems with a Set-Membership Description of Uncertainty*. PhD thesis, MIT, 1971.

Alexander Binder, Klaus-Robert Müller, and Motoaki Kawanabe. On taxonomies for multi-class image categorization. *International Journal of Computer Vision*, 99(3):281–301, 2012.

Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Workshop on Computational Learning Theory*, pages 144–152, 1992.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

Wei Chu and Zoubin Ghahramani. Gaussian processes for ordinal regression. *Journal of Machine Learning Research*, 6(Jul):1019–1041, 2005.

Wei Chu and S Sathiya Keerthi. New approaches to support vector ordinal regression. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 145–152. ACM, 2005.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3): 273–297, 1995.

Corinna Cortes, Giulia DeSalvo, and Mehryar Mohri. Boosting with abstention. In *Advances in Neural Information Processing Systems*, pages 1660–1668, 2016.

Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research*, 2:265–292, 2002.

George Dantzig. *Linear programming and extensions*. RAND Corporation, 1963.

George B Dantzig. Programming in a linear structure. *Washington, DC*, 1948.

Erick Delage and Yinyu Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations research*, 58(3):595–612, 2010.

Naiyang Deng, Yingjie Tian, and Chunhua Zhang. *Support vector machines: optimization based theory, algorithms, and extensions*. CRC press, 2012.

Ürün Doğan, Tobias Glasmachers, and Christian Igel. A unified view on multi-class support vector classification. *Journal of Machine Learning Research*, 17(45):1–32, 2016.

Rizal Fathony, Anqi Liu, Kaiser Asif, and Brian Ziebart. Adversarial multiclass classification: A risk minimization perspective. In *Advances in Neural Information Processing Systems*, pages 559–567, 2016.

Rizal Fathony, Mohammad Ali Bashiri, and Brian Ziebart. Adversarial surrogate losses for ordinal regression. In *Advances in Neural Information Processing Systems*, pages 563–573, 2017.

Rizal Fathony, Sima Behpour, Xinhua Zhang, and Brian Ziebart. Efficient and consistent adversarial bipartite matching. In *International Conference on Machine Learning*, pages 1456–1465, 2018.

Yoav Freund and Robert 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.

Yves Grandvalet, Alain Rakotomamonjy, Joseph Keshet, and Stephane Canu. Support vector machines with a reject option. In *Advances in Neural Information Processing Systems*, pages 537–544, 2009.

Peter D. Grünwald and A. Phillip Dawid. Game theory, maximum entropy, minimum discrepancy, and robust Bayesian decision theory. *Annals of Statistics*, 32:1367–1433, 2004.

Klaus-Uwe Hoffgen, Hans-Ulrich Simon, and Kevin S Vanhorn. Robust trainability of single neurons. *Journal of Computer and System Sciences*, 50(1):114–125, 1995.

Christian Igel, Verena Heidrich-Meisner, and Tobias Glasmachers. Shark. *Journal of Machine Learning Research*, 9:993–996, 2008.

Narendra Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, pages 302–311. ACM, 1984.

Yoonkyung Lee, Yi Lin, and Grace Wahba. 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, 2004.

Ling Li and Hsuan-Tien Lin. Ordinal regression by extended binary classification. *Advances in Neural Information Processing Systems*, 19:865, 2007.

M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

Hsuan-Tien Lin. *From ordinal ranking to binary classification.* PhD thesis, California Institute of Technology, 2008.

Hsuan-Tien Lin. Reduction from cost-sensitive multiclass classification to one-versus-one binary classification. In *Proceedings of the Sixth Asian Conference on Machine Learning*, pages 371–386, 2014.

Hsuan-Tien Lin and Ling Li. Large-margin thresholded ensembles for ordinal regression: Theory and practice. In *International Conference on Algorithmic Learning Theory*, pages 319–333. Springer, 2006.

Yi Lin. Support vector machines and the bayes rule in classification. *Data Mining and Knowledge Discovery*, 6(3):259–275, 2002.

Yufeng Liu. Fisher consistency of multicategory support vector machines. In *International Conference on Artificial Intelligence and Statistics*, pages 291–298, 2007.

Peter McCullagh and John A Nelder. *Generalized linear models*, volume 37. CRC press, 1989.

Fabian Pedregosa, Francis Bach, and Alexandre Gramfort. On the consistency of ordinal regression methods. *Journal of Machine Learning Research*, 18(55):1–35, 2017.

Harish G Ramaswamy and Shivani Agarwal. Classification calibration dimension for general multiclass losses. In *Advances in Neural Information Processing Systems*, pages 2078–2086, 2012.

Harish G Ramaswamy and Shivani Agarwal. Convex calibration dimension for multiclass loss matrices. *The Journal of Machine Learning Research*, 17(1):397–441, 2016.

Harish G Ramaswamy, Ambuj Tewari, Shivani Agarwal, et al. Consistent algorithms for multiclass classification with an abstain option. *Electronic Journal of Statistics*, 12(1): 530–554, 2018.

Jason D. M. Rennie and Nathan Srebro. Loss functions for preference levels: Regression with discrete ordered labels. In *Proceedings of the IJCAI Multidisciplinary Workshop on Advances in Preference Handling*, pages 180–186, 2005.

Ralph Tyrell Rockafellar. *Convex Analysis*. Princeton Mathematics Series. Princeton University Press, Princeton, NJ, 1970.

Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: Primal estimated sub-gradient solver for SVM. *Mathematical Programming*, 127(1):3–30, 2011.

Amnon Shashua and Anat Levin. Ranking with large margin principle: Two approaches. In *Advances in Neural Information Processing Systems 15*, pages 961–968. MIT Press, 2003.

M. Sion. On general minimax theorems. *Pacific Journal of Mathematics*, 8(1):171–176, 1958.

Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Springer Publishing Company, Incorporated, 1st edition, 2008. ISBN 0387772413.

Ambuj Tewari and Peter L Bartlett. On the consistency of multiclass classification methods. *The Journal of Machine Learning Research*, 8:1007–1025, 2007.

Han-Hsing Tu and Hsuan-Tien Lin. One-sided support vector regression for multiclass cost-sensitive classification. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 1095–1102, 2010.

Vladimir Vapnik. Principles of risk minimization for learning theory. In *Advances in Neural Information Processing Systems*, pages 831–838, 1992.

John Von Neumann and Oskar Morgenstern. Theory of games and economic behavior. *Bulletin of the American Mathematical Society*, 51(7):498–504, 1945.

Jason Weston, Chris Watkins, et al. Support vector machines for multi-class pattern recognition. In *ESANN*, volume 99, pages 219–224, 1999.