Performance-Aligned Learning Algorithms with Statistical Guarantees

by

Rizal Zaini Ahmad Fathony
B.A.S., Institute of Statistics, Jakarta, 2007
M.S., University of Illinois at Chicago, Chicago, 2014

Thesis submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Chicago, 2019

Chicago, Illinois

Defense Committee:
Brian D. Ziebart, Chair and Advisor
Bhaskar DasGupta
Xinhua Zhang
Lev Reyzin, Mathematics, Statistics, and Computer Science
Simon Lacoste-Julien, Université de Montréal

To Lia and Hana

ACKNOWLEDGMENTS

First and foremost, I want to express my sincere gratitude to my advisor, Prof. Brian Ziebart, for his continuous guidance and support during my Ph.D. study. I appreciate all his contributions in guiding me through the adventurous world of machine learning research. I would like to thank Prof. Xinhua Zhang for introducing and guiding me to the mathematical optimization research. I would like to also thank the rest of my thesis committee members: Prof. Bhaskar DasGupta, Prof. Lev Reyzin, and Prof. Simon Lacoste-Julien, for their comments and help on the thesis and defense.

I also want to thank my collaborators: Anqi Liu, Kaiser Asif, Mohammad Ali Bashiri, Ashkan Rezaei, Sima Behpour, and Wei Xing, for their contributions to get our works published. I thank my other labmates: Mathew Monfort, Xiangli Chen, Hong Wang, Jia Li, Sanket Gaurav, Zainab Al-Qurashi, Chris Schultz, and George Maratos, for the encouragement and feedback they provided.

Last but not the least, I would like to thank my wife, Lia Amelia, and my daughter, Hana Fathony, for their unconditional support through the ups and downs of my Ph.D. study.

RZAF

CONTRIBUTIONS OF AUTHOR

Chapter 2 presents a published manuscript (Fathony et al., 2018c), which is the extension of published conference papers (Fathony et al., 2016; Fathony et al., 2017). I was the primary author of the manuscript and the conference papers. I formulated and proved most of the theorems in the papers, constructed the model formulations, designed and implemented the optimization algorithms, and conducted most of the experiments under the guidance of my advisor, Prof. Brian D. Ziebart. Prof. Ziebart provided the general formulation of adversarial prediction (Definition 2.1). Prof. Xinhua Zhang provided guidance in formal mathematical proof. I wrote and revise the manuscript in close collaboration with Prof. Ziebart and Prof. Zhang. Anqi Liu contributed to the discussions about the methods and their consistency properties. Kaiser Asif, Anqi Liu, and Mohammad Ali Bashiri contributed in implementing the baselines and conducting the experiments for the baselines.

Chapter 3 presents a published paper (Fathony et al., 2018b) in which I was the primary author. I formulated and proved most of the theorems in the papers, constructed the model formulations, designed and implemented the optimization algorithms, and conducted most of the experiments under the guidance of Prof. Ziebart. Prof. Zhang provided guidance in the optimization algorithm. Ashkan Rezaei and Mohammad Ali Bashiri contributed in conducting the semantic role labelling experiments.

Chapter 4 presents a published paper (Fathony et al., 2018a) in which I was one of the primary authors. I formulated and derived the marginal distribution technique for efficiently

CONTRIBUTIONS OF AUTHOR (Continued)

solving the adversarial bipartite matching problem, designed and implemented the optimization algorithm, analyzed the consistency property, and conducted the marginal formulation experimental works under the guidance of Prof. Ziebart. Prof. Zhang provided guidance in the optimization algorithm and consistency property derivation. Sima Behpour contributed to the double-oracle version of the optimization technique (implementation and experiments) and designed the feature representation for the experiments.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

CHAPTER **PAGE**

TABLE OF CONTENTS (Continued)

CHAPTER **PAGE**

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (Continued)

SUMMARY

The goal of many prediction tasks in machine learning is to learn a prediction function that minimizes certain loss metrics (e.g., zero-one, ordinal, and cost-sensitive loss) or maximizes certain performance metrics (e.g., accuracy, precision, recall, F1-score, and ROC curve) on the testing dataset. Unfortunately, optimizing these metrics directly via empirical risk minimization is known to be intractable. In practice, convex surrogate losses over the desired metrics are needed in order to build efficient learning algorithms with the hope that optimizing the convex surrogates will indirectly optimize the original metrics given sufficient training data.

Probabilistic and large-margin approaches are two popular paradigms for constructing learning algorithms that differ in the way they construct convex surrogate losses. Probabilistic approaches construct prediction probability models and employ the logistic loss as the convex surrogate. Large-margin approaches aim to maximize the margin that separates correct predictions from incorrect ones and use the hinge loss for the convex surrogate construction. Both approaches have their own strengths and weaknesses. The probabilistic approaches enjoy the statistical guarantee of Fisher consistency, meaning it optimizes the desired performance/loss metric and produces Bayes optimal classifiers when they learn from any true distribution of data using a rich feature representation. The large-margin approaches enjoy the computational efficiency and also the flexibility of aligning the optimization algorithm with the desired performance/loss metrics. However, in many cases, probabilistic approaches do not have a mechanism to easily incorporate customized performance/loss metrics into their learning process, whereas

large-margin models do not have Fisher consistency guarantees in general (except for the binary classification case and a few multiclass formulations). This motivates the search for new approaches that overcome the weaknesses of the probabilistic and large-margin methods.

This thesis addresses the challenges above by constructing new learning algorithms that simultaneously satisfy the desired properties of: (1) *aligning with the learning objective by incorporating customized performance/loss metrics into the learning process*; (2) *providing the statistical guarantee of Fisher consistency*; (3) *enjoying computational efficiency*; and (4) *performing competitively in practice*. Our approach for constructing the learning algorithms is based on the robust adversarial formulation, i.e., by focusing on answering the question: *"what predictor best maximizes the performance metric (or minimizes the loss metric) in the worst case given the statistical summaries of the empirical distributions?"* We focus on two different areas of machine learning: general multiclass classification and structured prediction. In both areas, we demonstrate the theoretical and practical benefit of our methods.

CHAPTER 1

INTRODUCTION

1.1 Machine Learning Tasks

Machine learning has been successfully implemented in many real world applications in different areas. From understanding the meaning of text (Cohn and Blunsom, 2005; Chen et al., 2017; Tang et al., 2016) and translating languages (Vaswani et al., 2018; Lample et al., 2018) in the domain of natural language processing or classifying and segmenting images in the domain of computer vision (Hu et al., 2018; Lin et al., 2018; Chen et al., 2018), to predicting diseases (Purushotham et al., 2018; Menegola et al., 2017; Rakhlin et al., 2018) and conducting science experiments (Regier et al., 2015; Albertsson et al., 2018), machine learning provides tools that help solve these tasks. In many of these prediction tasks, the goal of machine learning algorithms is to learn a prediction function that minimizes certain loss metrics (e.g., zero-one, ordinal, and cost-sensitive loss) or maximizes certain performance metrics (e.g., accuracy, precision, recall, F1-score, and ROC curve) on the testing dataset.

The accuracy performance metric (or the zero-one loss metric) is the most widely used metric for classification tasks where a learning algorithm needs to choose a prediction from a finite set of class labels. Some of the example of this classification task are classifying images and predicting whether a patient has a particular disease. In some tasks where the class label has an inherent order (e.g., `poor`, `fair`, `good`, `very good`, and `excellent` labels), ordinal loss

metrics such as the absolute and square losses are often used (Pedregosa et al., 2017). Many application tasks where the number of samples in some classes are imbalance (for example, the task of predicting diseases in health and medicine areas, and several tasks in information retrieval area), the accuracy metric is unsatisfactory. More appropriate metrics such as the F-score metric (which is based on precision and recall metrics), the area under the ROC curve metric, or cost-sensitive loss metrics are often preferred to evaluate the performance of learning algorithms in these tasks (He and Ma, 2013).

One of the popular principles in designing learning algorithms is empirical risk minimization (ERM) (Vapnik, 1992). The ERM framework suggests finding a classifier that minimizes the risk with respect to the training data, which is the expected value of the loss metric (or performance metric) produced by a classifier. In practice, it is common to use a modification of the ERM framework called the structured risk minimization (SRM) framework, which considers the balance between minimizing the risk and generalization error by penalizing the model complexity (Vapnik, 1992). Unfortunately, since most of the loss/performance metrics are discrete, non-convex, and non-continuous, optimizing these metrics directly via empirical/structural risk minimization is known to be intractable once the set of hypotheses is (parametrically) restricted (e.g., as a linear function of input features) (Hoffgen et al., 1995; Steinwart and Christmann, 2008). To avoid this intractability, many machine learning models employ convex surrogate losses over the desired metric in order to build efficient learning algorithms with the hope that optimizing the convex surrogate will indirectly optimize the original metric given sufficient training data.

1.2 Two Major Paradigms in Machine Learning

Many different machine learning models can be categorized based on the way they construct their convex surrogate loss. Among the most popular paradigms are probabilistic approaches and large-margin approaches. Probabilistic approaches construct predictive probability distributions and employ the logistic loss as the convex surrogate. Large-margin approaches aim to maximize the margin that separates correct predictions from the incorrect ones and use the hinge loss for the convex surrogate's construction.

Both paradigms have produced many machine learning models with different characteristics for many different tasks. In the most basic binary classification task, probabilistic and large-margin approaches produce two of the most widely used machine learning models, the logistic regression model (Cox, 1958) and the support vector machine (SVM) model (Cortes and Vapnik, 1995) respectively. The probabilistic approach extends naturally to the multiclass classification task as multinomial logistic regression (McCullagh and Nelder, 1989). On the contrary, there are many competing formulations of the large-margin approach for multiclass classification (Weston et al., 1999; Crammer and Singer, 2002; Lee et al., 2004; Doğan et al., 2016). In the ordinal classification task, many extensions from binary classification have been proposed for both probabilistic and large-margin approaches (Shashua and Levin, 2003; Chu and Keerthi, 2005; Rennie and Srebro, 2005). For more complex loss/performance metrics, the large margin approach is more commonly used with the introduction of SVM^{perf} (Joachims, 2005), which is an extension of the SVM algorithm that accepts custom performance metrics. Both probabilistic and large-margin approaches can also be combined with neural networks and deep learning

approaches where the convex surrogate losses produced by the probabilistic and large-margin approaches are used as the last layer (output unit) in the network architecture. In practice, though, the probabilistic approaches are more popular than the large-margin approach in neural network constructions (Goodfellow et al., 2016).

In the tasks of structured prediction and graphical models, where a learning algorithm needs to predict multiple variables simultaneously by considering the relationship among the variables, the probabilistic approach provides a tool to model the independence properties among the variables. This results in several probabilistic models, e.g., hidden Markov models (HMM) (Baum and Petrie, 1966) and conditional random fields (CRF) (Lafferty et al., 2001), that differ in the way they encode independence properties. On the other hand, the large-margin approach extends the formulation of multiclass SVM to structured prediction problems. Two main formulations in this approach are the maximum margin Markov network (M^3N) (Taskar et al., 2005a) and the structured support vector machine (SSVM) (Tsochantaridis et al., 2005).

1.3 Strengths and Weaknesses

The probabilistic and large-margin paradigms offer two different ways to construct convex surrogate losses for many prediction tasks. Each paradigm come with its own strengths and weaknesses. In the case of multiclass classification, for example, the probabilistic approach (logistic regression) enjoys the statistical guarantee of Fisher consistency, meaning it optimizes the accuracy metric and produces Bayes optimal classifiers when they learn from any true distribution of data using a rich feature representation (Bartlett et al., 2006). On the other hand, the large-margin approach (SVM) enjoys computational efficiency via the kernel trick

and dual parameter sparsity (Cortes and Vapnik, 1995). However, many formulations of the large-margin approach suffer from Fisher consistency issues (Tewari and Bartlett, 2007; Liu, 2007; Doğan et al., 2016), while the probabilistic approach does not have dual parameter sparsity (Zhu and Hastie, 2002).

When generalized to structured prediction, probabilistic methods such as conditional random field (CRF) (Lafferty et al., 2001) capture probabilistic structures in the model (which translates to Fisher consistency guarantees), with the downside that the computation of the normalization term may be intractable. The other weakness of probabilistic methods is that they do not have an easy mechanism to incorporate customized performance/loss metrics into their learning process, which is important in many structured prediction settings. Large-margin models like structured SVM (SSVM) (Tsochantaridis et al., 2005) have the flexibility to align with the desired performance/loss metric (by incorporating it into the learning process), but the Fisher consistency property is not guaranteed.

1.4 Combining the Best of Both Worlds

The weaknesses of the probabilistic and large-margin paradigms motivate us to search for a new approach that overcome these problems. In particular, we are interested in combining the performance-aligned property of the large-margin methods and the statistical guarantee of the probabilistic models.

This thesis addresses these challenges by constructing new learning algorithms that simultaneously satisfy the desired properties of: (1) *aligning with the learning objective by incorporating customized performance/loss metrics into the learning process*; (2) *providing the statistical guar-*

antee of Fisher consistency; (3) *enjoying computational efficiency*; and (4) *performing competitively in practice*. Our approach for constructing the learning algorithms is based on the robust adversarial formulation (Topsøe, 1979; Grünwald and Dawid, 2004; Delage and Ye, 2010; Asif et al., 2015), i.e., by focusing on answering the question: *"what predictor best maximizes the performance metric (or minimizes the loss metric) in the worst case given the statistical summaries of the empirical distributions?"* In this thesis, we show that this predictor satisfies the desired properties of learning algorithms above. We focus on two different areas of machine learning: general multiclass classification and structured prediction.

1.5 Thesis Outline

To demonstrate the formulations, theoretical properties, optimization algorithms and practical benefits of our learning algorithms, we divide our contributions into three chapters:

- Chapter 2: Performance-Aligned Surrogate Losses for General Multiclass Classification.
- Chapter 3: Performance-Aligned Adversarial Graphical Models.
- Chapter 4: Adversarial Bipartite Matching in Graphs.

The first and the last chapters serve as the introduction to the topic of this thesis and the overall conclusion of the thesis respectively.

Performance-Aligned Surrogate Losses for General Multiclass Classification

Our first focus is in general multiclass classification, i.e., multiclass classification with an arbitrary loss metric. We start with the adversarial learning formulation for general multiclass classification problem that can be aligned with the desired loss metric. We then take a dual

perspective of the formulation and view it as a surrogate loss over the desired loss metric. We derive compact forms of surrogate losses for several metrics (including the zero-one, absolute, squared, and abstention loss metric), and construct efficient algorithms to compute the surrogate losses. For theoretical justification, we show that our surrogate losses satisfy the statistical guarantee of Fisher consistency. We develop efficient optimization algorithms as well as a way to incorporate richer feature spaces via the kernel trick. Finally, we demonstrate the effectiveness of our models in several prediction tasks.

Performance-Aligned Adversarial Graphical Models

We next move to structured prediction problems—specifically—prediction tasks in which the relation among predicted variables are represented in a graph. We compare our approach with CRF and structured SVM (SSVM) predictors. The CRF has the advantage of Fisher consistency but cannot be easily aligned with custom performance/loss metrics, whereas the SSVM can easily align with the metric but without consistency guarantee. We extend our adversarial formulation for multiclass classification to the graphical model problems. To avoid the need for computing over exponentially many possible label values in structured prediction, we formulate our optimization in terms of the node and edge marginal distributions in the graphical structure. This results in a learning algorithm that has runtime complexity competitive with the CRF and SSVM. Our formulation can be aligned with any loss metrics that additively decompose over the nodes in the graph. We show that our approach enjoys the Fisher consistency guarantee as in the CRF. We then demonstrate the benefit of our methods in two different prediction tasks: the first one is over chain structures, and the second one is over tree structures.

Adversarial Bipartite Matching in Graphs

The bipartite matching task is a special case of graphical model where we have two equally-sized sets of nodes, and aim to model the one-to-one correspondence between the nodes in the first set with the nodes in the second set. The task is usually formulated as modelling permutations that corresponds to the matching assignments. The CRF model, though possessing desirable consistency properties, is known to be intractable in this task due to the need of computing the matrix permanent (a #P-hard problem) in the computation of its normalization term. Consequently, for a modestly size problem, one needs to employ approximations in order to apply the method. The SSVM, on the other hand, has a polynomial time algorithm for computing the most violated constraints in its optimization, leading to an overall polynomial time complexity. However, as in the standard graphical model, the consistency of SSVM is not guaranteed. We focus on optimizing the Hamming loss metric in our bipartite matching formulation. Using a similar marginal formulation technique, we formulate an efficient learning algorithm for this task. We also show the Fisher consistency of our method. We demonstrate the benefit of our consistent method compared to the SSVM model in several bipartite matching tasks. In some of some of these prediction tasks, running the CRF is impractical due the size of the problems.

CHAPTER 2

PERFORMANCE-ALIGNED SURROGATE LOSSES FOR GENERAL MULTICLASS CLASSIFICATION

(Parts of this chapter were previously published as "Adversarial Multiclass Classification: A Risk Minimization Perspective" (Fathony et al., 2016) in the Advances in Neural Information Processing Systems 29 (NIPS 2016), as "Adversarial Surrogate Losses for Ordinal Regression" (Fathony et al., 2017) in the Advances in Neural Information Processing Systems 30 (NIPS 2017), and as "Consistent Robust Adversarial Prediction for General Multiclass Classification" (Fathony et al., 2018c) in arXiv preprint arXiv:1812.07526.)

2.1 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 or 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 once the set of hypotheses is (parametrically) restricted (e.g., as a linear function of input features) (Hoffgen et al., 1995; 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 (Cox, 1958), and the hinge loss, which is minimized by the support vector machine (SVM) (Boser et al., 1992; Cortes and Vapnik, 1995). Both 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.

2.2 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.

2.2.1 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, defined as: $\text{loss}^{\text{sq}}(\hat{y}, y) = (\hat{y} - y)^2$, 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 α is incurred whenever the predictor chooses to use the abstain option.

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

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

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).

2.2.2 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 θ that minimize the empirical risk,

$$\underset{\theta}{\text{argmin}} \, \mathbb{E}_{\mathbf{X}, Y \sim \tilde{P}} \left[\text{loss} \left(\hat{y}_\theta(\mathbf{X}), Y \right) \right] + \lambda ||\theta||, \tag{2.1}$$

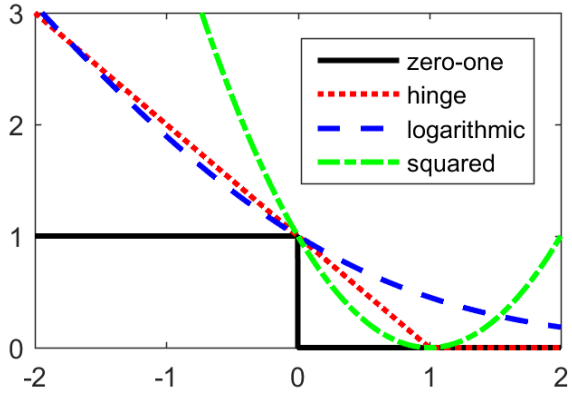


Figure 2. Convex surrogates for the zero-one loss.

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).

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

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

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. A technique to characterize the Fisher consistency of surrogate losses for the multiclass zero-one loss metric has been developed using the "classification calibration" concept (Tewari and Bartlett, 2007), which then been extended to general multiclass loss metrics (Ramaswamy and Agarwal, 2012; Ramaswamy and Agarwal, 2016).

2.2.3 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.

2.2.3.1 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 (Weston et al., 1999), which incorporates the sum of hinge losses for all alternative labels,

$$\text{loss}_{\text{WW}}(\mathbf{x}, y) = \sum_{j \neq y} [1 + (f_j(\mathbf{x}) - f_y(\mathbf{x}))]_+; \tag{2.2}$$

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

$$\text{loss}_{\text{CS}}(\mathbf{x}, y) = \max_{j \neq y} [1 + (f_j(\mathbf{x}) - f_y(\mathbf{x}))]_+ ; \text{ and} \tag{2.3}$$

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

$$\text{loss}_{\text{LLW}}(\mathbf{x}, y) = \sum_{j \neq y} [1 + f_j(\mathbf{x})]_+ , \tag{2.4}$$

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

The former two models (the 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. The 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, the LLW's use of an absolute margin in the loss (rather than the relative margin of the WW and CS) often causes it to perform poorly for datasets with low dimensional feature spaces (Doğan et al., 2016). 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.

2.2.3.2 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.

   Reduction framework is a technique to convert ordinal regression problems to binary classification problems by extending training examples (Li and Lin, 2007). 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; Lin, 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, cost-sensitive classification can also be reduced to one-sided regression (OSR)

problem, which can be viewed as an extension of the one-versus-all (OVA) technique (Tu and Lin, 2010).

In terms of Fisher consistency, many surrogate losses for ordinal classification enjoy this theoretical property. For example, the *all thresholds* methods are Fisher consistent provided that the base binary surrogate losses they use are convex with differentiability and a negative derivative at zero (Pedregosa et al., 2017).

2.2.3.3 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 α. In the literature, this type of prediction setting is also called "*classification with reject option*".

Most of the early studies on classification with abstention focused on the binary prediction case. For example, a consistent surrogate loss based on the SVM's hinge loss for binary classification with abstention where the value of α is restricted to the interval $[0, \frac{1}{2}]$ (Bartlett and Wegkamp, 2008), which then been extended to the case where the abstention penalty between the positive class α_+ and negative class α_- is non-symmetric (Grandvalet et al., 2009). Boosting algorithm (Freund and Schapire, 1997) can also be modified to incorporate the abstention setting into the prediction (Cortes et al., 2016). The method 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 multiclass classification setting, binary hinge loss can be extended to the case of multiclass classification with abstention (Ramaswamy et al., 2018). The study extended the definition of SVM's one-versus-all (OVA) and Crammer-Singer (CS) models to incorporate the abstention penalty. It 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.

2.3 Adversarial Prediction Formulation for Multiclass Classification

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 2.1. *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:*

$$\min_{\hat{P}(\hat{Y}|\mathbf{X})} \max_{\check{P}(\check{Y}|\mathbf{X})} \mathbb{E}_{\mathbf{X}\sim\tilde{P};\hat{Y}|\mathbf{X}\sim\hat{P};\check{Y}|\mathbf{X}\sim\check{P}} \left[loss(\hat{Y}, \check{Y}) \right] \tag{2.5}$$

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

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 2.1.

Theorem 2.1. *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:*

$$\min_{\theta} \mathbb{E}_{\mathbf{X},Y\sim\tilde{P}} \left[\max_{\check{P}(\check{Y}|\mathbf{X})} \min_{\hat{P}(\hat{Y}|\mathbf{X})} \mathbb{E}_{\hat{Y}|\mathbf{X}\sim\hat{P};\check{Y}|\mathbf{X}\sim\check{P}} \left[loss(\hat{Y},\check{Y}) + \theta^{\intercal} \left(\phi(\mathbf{X},\check{Y}) - \phi(\mathbf{X},Y) \right) \right] \right], \tag{2.6}$$

where θ is the Lagrange dual variable for the moment matching constraints.

Proof.

$$\min_{\hat{P}(\hat{Y}|\mathbf{X})} \max_{\check{P}(\check{Y}|\mathbf{X})} \mathbb{E}_{\mathbf{X}\sim\tilde{P};\hat{Y}|\mathbf{X}\sim\hat{P};\check{Y}|\mathbf{X}\sim\check{P}} \left[\text{loss}(\hat{Y},\check{Y}) \right] \tag{2.7}$$

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

$$\overset{(a)}{=} \max_{\check{P}(\check{Y}|\mathbf{X})} \min_{\hat{P}(\hat{Y}|\mathbf{X})} \mathbb{E}_{\mathbf{X}\sim\tilde{P};\hat{Y}|\mathbf{X}\sim\hat{P};\check{Y}|\mathbf{X}\sim\check{P}} \left[\text{loss}(\hat{Y},\check{Y}) \right] \tag{2.8}$$

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

$$\overset{(b)}{=} \max_{\check{P}(\check{Y}|\mathbf{X})} \min_{\theta} \min_{\hat{P}(\hat{Y}|\mathbf{X})} \mathbb{E}_{\mathbf{X},Y\sim\tilde{P};\hat{Y}|\mathbf{X}\sim\hat{P};\check{Y}|\mathbf{X}\sim\check{P}} \left[\text{loss}(\hat{Y},\check{Y}) + \theta^{\intercal} \left(\phi(\mathbf{X},\check{Y}) - \phi(\mathbf{X},Y) \right) \right] \tag{2.9}$$

$$\overset{(c)}{=} \min_{\theta} \max_{\check{P}(\check{Y}|\mathbf{X})} \min_{\hat{P}(\hat{Y}|\mathbf{X})} \mathbb{E}_{\mathbf{X},Y\sim\tilde{P};\hat{Y}|\mathbf{X}\sim\hat{P};\check{Y}|\mathbf{X}\sim\check{P}} \left[\text{loss}(\hat{Y},\check{Y}) + \theta^{\intercal} \left(\phi(\mathbf{X},\check{Y}) - \phi(\mathbf{X},Y) \right) \right] \tag{2.10}$$

$$\overset{(d)}{=} \min_{\theta} \mathbb{E}_{\mathbf{X},Y\sim\tilde{P}} \left[\max_{\check{P}(\check{Y}|\mathbf{X})} \min_{\hat{P}(\hat{Y}|\mathbf{X})} \mathbb{E}_{\hat{Y}|\mathbf{X}\sim\hat{P};\check{Y}|\mathbf{X}\sim\check{P}} \left[\text{loss}(\hat{Y},\check{Y}) + \theta^{\intercal} \left(\phi(\mathbf{X},\check{Y}) - \phi(\mathbf{X},Y) \right) \right] \right]. \tag{2.11}$$

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 θ 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 θ is $\mathbb{R}^m$. The objective is concave on $\check{P}(\check{Y}|\mathbf{X})$ for all θ (a non-negative linear combination of minimums of affine functions is concave), while it is convex on θ 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 θ.

(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 (Equation (2.6)) possesses the important property of being a **convex** optimization problem in θ. The objective of Equation (2.6) 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 θ, 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 θ 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 θ. 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 (Equation (2.5)) are non-convex and non-continuous.

2.4 Adversarial Surrogate Losses

Despite the different motivations between our adversarial prediction framework and the empirical risk minimization framework, the dual optimization formulation (Equation (2.6)) resembles a risk minimization problem with the surrogate loss defined as:

$$AL(\mathbf{x}, y, \theta) = \max_{\check{P}(\check{Y}|\mathbf{x})} \min_{\hat{P}(\hat{Y}|\mathbf{x})} \mathbb{E}_{\hat{Y}|\mathbf{x}\sim\hat{P};\check{Y}|\mathbf{x}\sim\check{P}} \left[\text{loss}(\hat{Y}, \check{Y}) + \theta^{\intercal} \left(\phi(\mathbf{x}, \check{Y}) - \phi(\mathbf{x}, y)\right)\right]. \tag{2.12}$$

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:

$$AL(\mathbf{f}, y) = \max_{\mathbf{q}\in\Delta} \min_{\mathbf{p}\in\Delta} \mathbf{p}^\intercal \mathbf{L}\mathbf{q} + \mathbf{f}^\intercal \mathbf{q} - f_y, \tag{2.13}$$

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

$$\begin{aligned} AL(\mathbf{f}, y) = \max_{\mathbf{q}, v}\ & v + \mathbf{f}^\intercal \mathbf{q} - f_y \\ \text{s.t.: } & \mathbf{L}_{(i,:)}\mathbf{q} \geq v \quad \forall i \in [k] \\ & q_i \geq 0 \qquad \forall i \in [k] \\ & \mathbf{q}^\intercal \mathbf{1} = 1, \end{aligned} \tag{2.14}$$

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.

2.4.1 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 Equation (2.14) for the case where $\mathbf{L}$ is the zero-one loss matrix. Since the objective in Equation (2.14) 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 (Rockafellar, 1970, Theorem 32.2). 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 2.2). Our strategy is to first characterize the extreme points of $\mathbb{C}$ that may possibly solve Equation (2.14), and then the evaluation of adversarial loss (AL) becomes equivalent to finding an extreme point that maximizes the objective in Equation (2.14).

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

$$\mathbb{C} = \left\{ \begin{bmatrix}\mathbf{q}\\ v\end{bmatrix} \;\middle|\; \mathbf{A}\begin{bmatrix}\mathbf{q}\\ v\end{bmatrix} \geq \mathbf{b}, \quad \text{where} \quad \mathbf{A} = \begin{bmatrix}\mathbf{L} & -\mathbf{1}\\ \mathbf{I} & \mathbf{0}\\ \mathbf{1}^\intercal & 0\\ -\mathbf{1}^\intercal & 0\end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix}\mathbf{0}\\ \mathbf{0}\\ 1\\ -1\end{bmatrix} \right\}. \tag{2.15}$$

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:

$$\begin{array}{l} \\ \text{1st block} \\ \\ \\ \\ \hdashline \\ \\ \\ \text{2nd block} \\ \\ \hdashline \\ \\ \text{3rd block} \end{array} \left[\begin{array}{ccccc} 0 & 1 & 1 & 1 & -1 \\ 1 & 0 & 1 & 1 & -1 \\ 1 & 1 & 0 & 1 & -1 \\ 1 & 1 & 1 & 0 & -1 \\ \hdashline 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \hdashline 1 & 1 & 1 & 1 & 0 \\ -1 & -1 & -1 & -1 & 0 \end{array}\right] \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ v \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ -1 \end{bmatrix}. \tag{2.16}$$

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 Equation (2.14), 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 2.1 (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 Equation (2.14).

Proposition 2.2. *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 Equation* (2.14).

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 Equation (2.14) 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 Equation (2.15). 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 $\mathbb{C}$.[1] By analyzing ext $\mathbb{C}$ in the case of multiclass zero-one classification, we simplify the adversarial surrogate loss (Equation (2.14)) as stated in the following Theorem 2.2.

---

[1]Indeed, it is the bona fide extreme point set of $\mathbb{C}$ under the standard definition which does not require compactness (Rockafellar, 1970, Section 18). But the guarantee of attaining optimality at an extreme point does require boundedness.

Theorem 2.2. *The model parameter θ for multiclass zero-one adversarial classification is equivalently obtained from empirical risk minimization under the adversarial zero-one loss function:*

$$AL^{0\text{-}1}(\mathbf{f}, y) = \max_{S \subseteq [k],\, S \neq \emptyset} \frac{\sum_{i \in S} f_i + |S| - 1}{|S|} - f_y, \tag{2.17}$$

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"[1]

$$D = \left\{ \begin{bmatrix} \mathbf{q} \\ v \end{bmatrix} = \frac{1}{|S|} \begin{bmatrix} \sum_{i \in S} \mathbf{e}_i \\ |S| - 1 \end{bmatrix} \;\middle|\; \emptyset \neq S \subseteq [k] \right\}, \tag{2.18}$$

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

Given $\mathbf{c} \in \text{ext}\, \mathbb{C}$, suppose the set of rows that $\mathbf{A}_\mathbf{c}$ 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 $\mathbf{A}$. We first observe that R and T must be disjoint because if $i \in R \cap T$, then $q_i = 0$ and $v = \mathbf{L}_{(i,:)}\mathbf{q} =$

---

[1]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.

$\sum_{j \neq i} q_j = 1 - q_i = 1$. 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 yield $q_j = 1/|R|$ 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 Equation (2.17), can be equivalently written as:

$$\text{AL}^{\text{0-1}}(\mathbf{f}, y) = \max_{S \subseteq [k],\, S \neq \emptyset} \frac{\sum_{i \in S} \psi_{i,y} + |S| - 1}{|S|}. \tag{2.19}$$

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 ψ 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 Equation (2.12).

For three classes, the loss function has seven facets as shown in Figure 4a. Figure 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

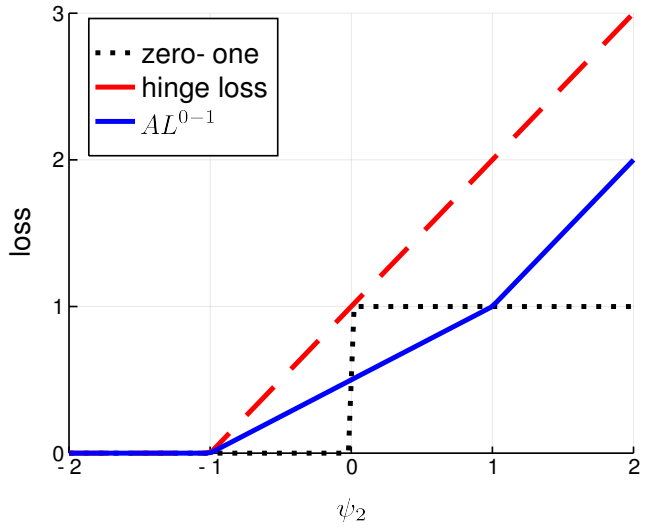


Figure 3. $\text{AL}^{\text{0-1}}$ evaluated over the space of potential differences ($\psi_{i,y} = f_i - f_y$; and $\psi_{i,i} = 0$) for binary prediction tasks when the true label is $y = 1$.

pitfall pointed out by (Doğan et al., 2016) which states that surrogate losses based on the absolute margin (rather than relative margin) may suffer from low performance for datasets with low dimensional 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 sorted order, for some value of j. Therefore, to compute $\text{AL}^{\text{0-1}}$, we can incrementally add the label in the sorted order to the set S^* until adding an additional label would decrease the value of the loss. 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.

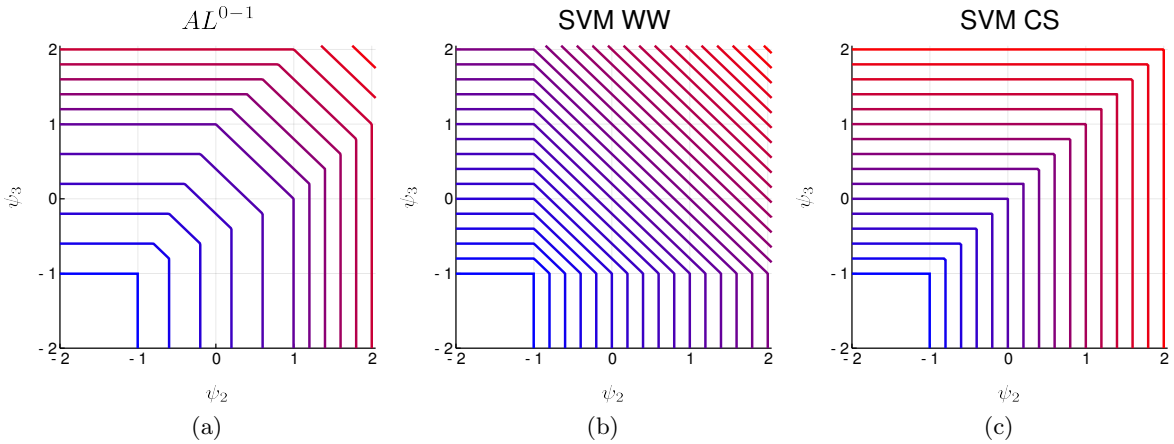


Figure 4. Loss function contour plots over the space of potential differences for the prediction task with three classes when the true label is $y = 1$ under $\text{AL}^{0\text{-}1}$ (a), the WW loss (b), and the CS loss (c). (Note that ψ_i in the plots refers to $\psi_{i,y} = f_i - f_y$; and $\psi_{i,i}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

Theorem 2.3. *The algorithm for computing $AL^{0\text{-}1}$ above is optimal.*

Proof. To calculate the set S^* that maximize $\text{AL}^{0\text{-}1}$ given the potentials of for all labels sorted in non-increasing order, the algorithm starts with the empty set $S = \emptyset$, it then adds labels to S in sorted order until adding a label would decrease the value of $\frac{\sum_{i \in S} f_i + |S| - 1}{|S|}$. If the set that maximizes $\text{AL}^{0\text{-}1}$ has j elements, it must contain the j largest potentials, otherwise we can swap the potentials that are not in the j largest potentials list with the potentials in the list and get a larger value. We are now left to prove that adding more potentials will not increase the value of the loss.

Let f_i denote the potentials sorted in non-increasing order, i.e. $f_1 \geq f_2 \geq \cdots \geq f_k$, and let j be the size of the set S^*, hence $\frac{\sum_{i \in S^*} f_i + |S^*| - 1}{|S^*|} = \frac{\sum_{i=1}^{j} f_i + j - 1}{j}$. We aim to prove that

$\frac{\sum_{i=1}^{j} f_i+j-1}{j} \geq \frac{\sum_{i=1}^{j+l} f_i+j+l-1}{j+l}$ for any $l = \{1, \ldots, k-j\}$. From the construction of the algorithm we know that it is true for $l = 1$, i.e.,:

$$\frac{\sum_{i=1}^{j} f_i + j - 1}{j} \geq \frac{\sum_{i=1}^{j+1} f_i + j}{j+1} \tag{2.20}$$

$$(j+1)\left(\sum_{i=1}^{j} f_i + j - 1\right) \geq j\left(\sum_{i=1}^{j+1} f_i + j\right) \tag{2.21}$$

$$j\sum_{i=1}^{j} f_i + j^2 - j + \sum_{i=1}^{j} f_i + j - 1 \geq j\sum_{i=1}^{j} f_i + jf_{j+1} + j^2 \tag{2.22}$$

$$\sum_{i=1}^{j} f_i - 1 \geq jf_{j+1}. \tag{2.23}$$

Since the potentials are sorted in non-increasing order, then for any $l = \{1, \ldots, k-j\}$:

$$l\left(\sum_{i=1}^{j} f_i - 1\right) \geq ljf_{j+1} \geq j\sum_{i=1}^{l} f_{j+l} \tag{2.24}$$

$$l\sum_{i=1}^{j} f_i - l + j\sum_{i=1}^{j} f_i + j^2 + j(l-1) \geq j\sum_{i=1}^{l} f_{j+l} + j\sum_{i=1}^{j} f_i + j^2 + j(l-1) \tag{2.25}$$

$$j\sum_{i=1}^{j} f_i + j^2 - j + l\sum_{i=1}^{j} f_i + lk - l \geq j\sum_{i=1}^{j+l} f_i + j^2 + j(l-1) \tag{2.26}$$

$$(j+l)\left(\sum_{i=1}^{j} f_i + j - 1\right) \geq j\left(\sum_{i=1}^{j+l} f_i + j + l - 1\right) \tag{2.27}$$

$$\frac{\sum_{i=1}^{j} f_i + j - 1}{j} \geq \frac{\sum_{i=1}^{j+l} f_i + j + l - 1}{j+l}. \tag{2.28}$$

Therefore, we can conclude that the algorithm for computing $\text{AL}^{\text{0-1}}$ is optimal. □

2.4.2 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 Equation (2.14), where $\mathbf{L}$ is the absolute loss matrix (e.g., Figure 1b for a five-class ordinal classification). The constraints in Equation (2.14) 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.

$$\begin{array}{l} \\ \text{1st block} \\ \\ \\ \\ \hdashline \\ \\ \\ \text{2nd block} \\ \\ \hdashline \\ \\ \text{3rd block} \end{array} \begin{bmatrix} 0 & 1 & 2 & 3 & -1 \\ 1 & 0 & 1 & 2 & -1 \\ 2 & 1 & 0 & 1 & -1 \\ 3 & 2 & 1 & 0 & -1 \\ \hdashline 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \hdashline 1 & 1 & 1 & 1 & 0 \\ -1 & -1 & -1 & -1 & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ v \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ -1 \end{bmatrix}. \tag{2.29}$$

By analyzing the extreme points of $\mathbb{C}$, we define the adversarial surrogate loss for ordinal classification with absolute loss AL^{ord} as stated in Theorem 2.4.

Theorem 2.4. *An adversarial ordinal classification predictor with absolute loss is obtained by choosing parameters θ that minimize the empirical risk of the surrogate loss function:*

$$AL^{ord}(\mathbf{f}, y) = \max_{i,j \in [k]; i \leq j} \frac{f_i + f_j + j - i}{2} - f_y. \tag{2.30}$$

Proof. The AL^{ord} above corresponds to the set of "extreme points"

$$D = \left\{ \begin{bmatrix} \mathbf{q} \\ v \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \mathbf{e}_i + \mathbf{e}_j \\ j - i \end{bmatrix} \;\middle|\; i, j \in [k]; i \leq j \right\}. \tag{2.31}$$

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 $\frac{j-i}{2}$, where $i \leq j$.

Similar to the proof of Theorem 2.2, 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 $\mathbf{A}_{\mathbf{c}}$ 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 $\mathbf{A}_{\mathbf{c}}$ because $\text{rank}(\mathbf{A}_{\mathbf{c}}) = k + 1$. 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 AL^{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:

$$\max_{i,j\in[k];i\leq j} \frac{f_i + f_j + j - i}{2} - f_y = \max_{i,j\in[k]} \frac{f_i + f_j + j - i}{2} - f_y = \tfrac{1}{2}\max_i (f_i - i) + \tfrac{1}{2}\max_j (f_j + j) - f_y. \tag{2.32}$$

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

2.4.2.1 Feature Representations

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

$$\phi_{th}(\mathbf{x}, y) = \begin{pmatrix} y\mathbf{x} \\ I(y \leq 1) \\ I(y \leq 2) \\ \vdots \\ I(y \leq k-1) \end{pmatrix}; \text{ and } \quad \phi_{mc}(\mathbf{x}, y) = \begin{pmatrix} I(y=1)\mathbf{x} \\ I(y=2)\mathbf{x} \\ I(y=3)\mathbf{x} \\ \vdots \\ I(y=k)\mathbf{x} \end{pmatrix}. \tag{2.33}$$

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 η_1, η_2, ..., η_{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, ϕ_{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.

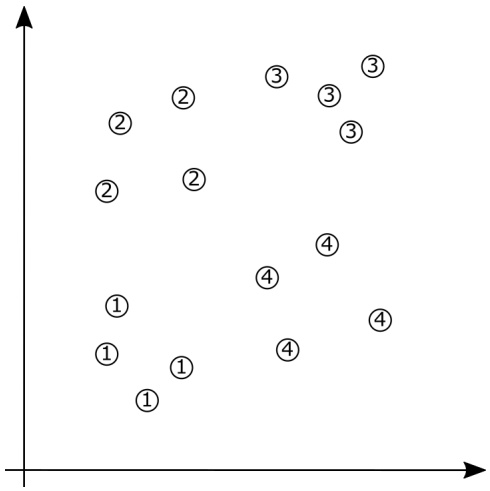


Figure 5. Example where multiple weight vectors are useful.

2.4.2.2 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 η_k associated with thresholds. Following Equa-

tion (2.32), the adversarial ordinal regression surrogate loss for this feature representation can be written as:

$$\mathrm{AL}^{\text{ord-th}}(\mathbf{x}, y) = \max_i \frac{i(\mathbf{w}\cdot\mathbf{x} - 1) + \sum_{l\geq i}\eta_l}{2} + \max_j \frac{j(\mathbf{w}\cdot\mathbf{x} + 1) + \sum_{l\geq j}\eta_l}{2} - y\mathbf{w}\cdot\mathbf{x} - \sum_{l\geq y}\eta_l. \quad (2.34)$$

This loss has a straight-forward interpretation in terms of the thresholded regression perspective, as shown in Figure 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.

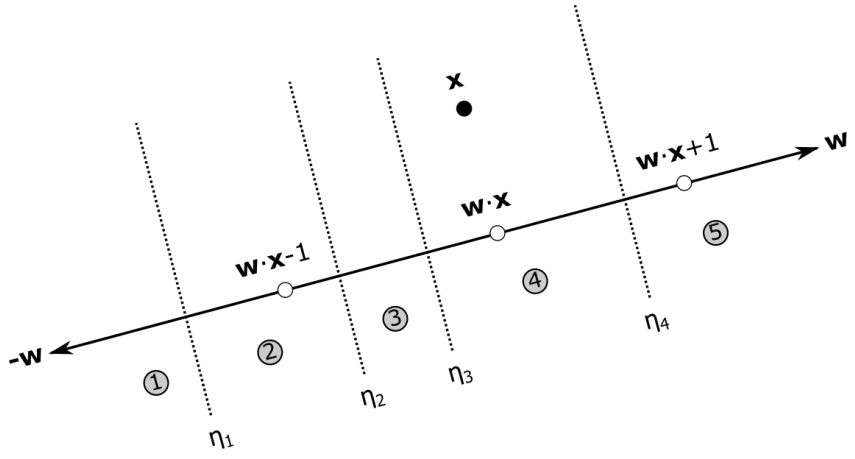


Figure 6. Surrogate loss calculation for datapoint $\mathbf{x}$ (projected to $\mathbf{w}\cdot\mathbf{x}$) with a label prediction of 4 for predictive purposes, the surrogate loss is instead obtained using potentials for the classes based on $\mathbf{w}\cdot\mathbf{x} - 1$ (label 2) and $\mathbf{w}\cdot\mathbf{x}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

2.4.2.3 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:

$$\mathrm{AL}^{\text{ord-mc}}(\mathbf{x}, y) = \max_{i,j \in [k]} \frac{\mathbf{w}_i \cdot \mathbf{x} + \mathbf{w}_j \cdot \mathbf{x} + j - i}{2} - \mathbf{w}_y \cdot \mathbf{x}. \tag{2.35}$$

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.

2.4.3 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: $\mathrm{loss}(\hat{y}, y) = (\hat{y} - y)^2$. The adversarial surrogate loss for ordinal classification using the squared loss metric is defined in Equation (2.14), where $\mathbf{L}$ is the squared loss matrix (e.g. Figure 1c for a five-class ordinal classification). The constraints in Equation (2.14) form a convex polytope $\mathbb{C}$. Below is an ex-

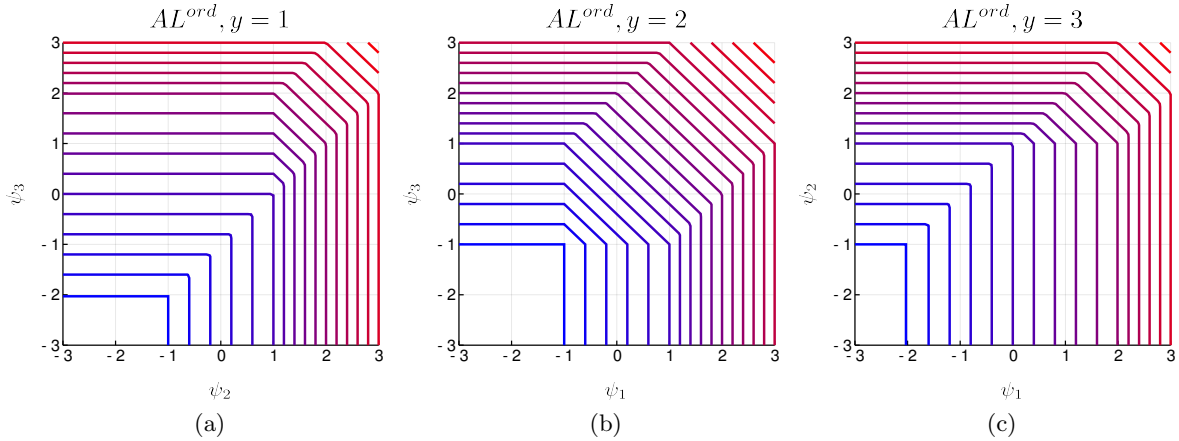


Figure 7. Loss function contour plots of $\mathrm{AL}^{\mathrm{ord}}$ over the space of potential differences $\psi_j \triangleq f_j - f_y$ for the prediction task with three classes when the true label is $y = 1$ (a), $y = 2$ (b), and $y = 3$ (c).

ample of the half-space representation of $\mathbb{C}$ for a four-class ordinal classification problem with squared loss metric.

$$\begin{array}{l} \\ \text{1st block} \\ \\ \\ \\ \hdashline \\ \\ \text{2nd block} \\ \\ \\ \hdashline \\ \text{3rd block} \\ \end{array} \begin{bmatrix} 0 & 1 & 4 & 9 & -1 \\ 1 & 0 & 1 & 4 & -1 \\ 4 & 1 & 0 & 1 & -1 \\ 9 & 4 & 1 & 0 & -1 \\ \hdashline 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \hdashline 1 & 1 & 1 & 1 & 0 \\ -1 & -1 & -1 & -1 & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ v \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ -1 \end{bmatrix}. \tag{2.36}$$

We define the adversarial surrogate loss for ordinal classification with squared loss $\mathrm{AL}^{\mathrm{sq}}$ as stated in Theorem 2.5.

Theorem 2.5. *An adversarial ordinal classification predictor with squared loss is obtained by choosing parameters θ that minimize the empirical risk of the surrogate loss function:*

$$AL^{sq}(\mathbf{f}, y) = \max \left\{ \max_{\substack{i,j,l \in [k] \\ i<l\leq j}} \frac{(2(j-l)+1)\left[f_i+(l-i)^2\right]+(2(l-i)-1)\left[f_j+(j-l)^2\right]}{2(j-i)},\ \max_i f_i \right\} - f_y. \tag{2.37}$$

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

$$D = \left\{ \begin{bmatrix} \mathbf{q} \\ v \end{bmatrix} = \tfrac{2(j-l)+1}{2(j-i)} \begin{bmatrix} \mathbf{e}_i \\ (l-i)^2 \end{bmatrix} + \tfrac{2(l-i)-1}{2(j-i)} \begin{bmatrix} \mathbf{e}_j \\ (j-l)^2 \end{bmatrix} \;\middle|\; \begin{matrix} i,j,l \in [k] \\ i<l\leq j \end{matrix} \right\} \cup \left\{ \begin{bmatrix} \mathbf{q} \\ v \end{bmatrix} = \begin{bmatrix} \mathbf{e}_i \\ 0 \end{bmatrix} \;\middle|\; i \in [k] \right\}. \tag{2.38}$$

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 2.2, 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 α, β, and γ. 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:

$$q_i = \frac{\mathbf{L}_{(l-1,j)} - \mathbf{L}_{(l,j)}}{\mathbf{L}_{(l,i)} - \mathbf{L}_{(l-1,i)} + \mathbf{L}_{(l-1,j)} - \mathbf{L}_{(l,j)}} \tag{2.39}$$

$$= \frac{(j-l+1)^2 - (j-l)^2}{(l-i)^2 - (l-1-i)^2 + (j-l+1)^2 - (j-l)^2} = \frac{2(j-l)+1}{2\,(j-i)}, \tag{2.40}$$

$$q_j = \frac{\mathbf{L}_{(l,i)} - \mathbf{L}_{(l-1,i)}}{\mathbf{L}_{(l,i)} - \mathbf{L}_{(l-1,i)} + \mathbf{L}_{(l-1,j)} - \mathbf{L}_{(l,j)}} \tag{2.41}$$

$$= \frac{(l-i)^2 - (l-1-i)^2}{(l-i)^2 - (l-1-i)^2 + (j-l+1)^2 - (j-l)^2} = \frac{2(l-i)-1}{2\,(j-i)}, \tag{2.42}$$

$$v = \frac{\left(\mathbf{L}_{(l-1,j)} - \mathbf{L}_{(l,j)}\right)\mathbf{L}_{(l,i)} + \left(\mathbf{L}_{(l,i)} - \mathbf{L}_{(l-1,i)}\right)\mathbf{L}_{(l,j)}}{\mathbf{L}_{(l,i)} - \mathbf{L}_{(l-1,i)} + \mathbf{L}_{(l-1,j)} - \mathbf{L}_{(l,j)}} \tag{2.43}$$

$$= \frac{(2(j-l)+1)\,(l-i)^2 + (2(l-i)-1)\,(j-l)^2}{2\,(j-i)}. \tag{2.44}$$

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

$$v = \tfrac{1}{z}\left\{(l-i)^2[(j-m)^2 - (j-l)^2] + (j-l)^2[(l-i)^2 - (m-i)^2]\right\}, \tag{2.45}$$

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:

$$w \triangleq \mathbf{L}_{(o,:)}\mathbf{q} = \tfrac{1}{z}\left\{(o-i)^2[(j-m)^2-(j-l)^2] + (j-o)^2[(l-i)^2-(m-i)^2]\right\}. \tag{2.46}$$

Then,

$$\begin{aligned} v - w =& \tfrac{1}{z}\big\{[(l-i)^2-(o-i)^2][(j-m)^2-(j-l)^2] \\ &- [(j-o)^2-(j-l)^2][(l-i)^2-(m-i)^2]\big\}. \end{aligned} \tag{2.47}$$

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

$$\frac{(l-i)^2-(o-i)^2}{(l-i)^2-(m-i)^2} > \frac{(j-o)^2-(j-l)^2}{(j-m)^2-(j-l)^2}. \tag{2.48}$$

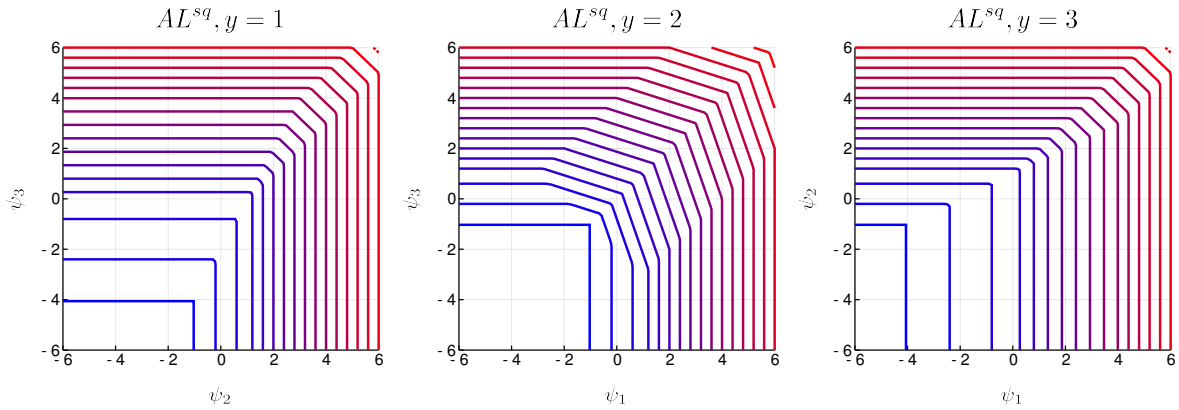
Thus, it contradicts with the requirement that $v \leq \mathbf{L}_{(o,:)}$.

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 $\mathrm{AL}^{\mathrm{sq}}$ 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 $\mathrm{AL}^{\mathrm{sq}}$ 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 $\mathrm{AL}^{\mathrm{sq}}$ is $\mathcal{O}(k^3)$. The loss surface of $\mathrm{AL}^{\mathrm{sq}}$ for the three classes classification is shown in Figure 8.



(a) (b) (c)

Figure 8. Loss function contour plots of $\mathrm{AL}^{\mathrm{sq}}$ 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).

2.4.4 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 α (for example, the weighted absolute loss: $\text{loss}(\hat{y}, y) = \alpha|\hat{y} - y|$). We next analyze in Theorem 2.6 the extreme points of the polytope formed by the the constraints in Equation (2.14) when $\mathbf{L}$ is the weighted multiclass loss metric.

Theorem 2.6. *Let $\mathbf{q}^*$, and v^* be the solution of the adversarial maximin (Equation* (2.14)*) with $\mathbf{L}$ as the loss matrix, then if the loss matrix is $\alpha\mathbf{L}$, the solution of (Equation* (2.14)*) 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 Equation (2.14) 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 αv^*. □

Using Theorem 2.6, we can derive the adversarial surrogate loss for weighted multiclass zero-one loss, absolute loss, and squared loss metrics as stated below.

Corollary 2.1. *An adversarial multiclass predictor with weighted zero-one loss is obtained by choosing the parameter θ that minimizes the empirical risk of the surrogate loss function:*

$$AL^{0\text{-}1\text{-}w}(\mathbf{f}, y, \alpha) = \max_{S \subseteq [k],\, S \neq \emptyset} \frac{\sum_{i \in S} f_i + \alpha\,(|S| - 1)}{|S|} - f_y. \tag{2.49}$$

Corollary 2.2. *An adversarial ordinal classification predictor with weighted absolute loss is obtained by choosing the parameter θ that minimizes the empirical risk of the surrogate loss function:*

$$AL^{ord\text{-}w}(\mathbf{f}, y, \alpha) = \max_{i,j \in [k]} \frac{f_i + f_j + \alpha\,(j - i)}{2} - f_y. \tag{2.50}$$

Corollary 2.3. *An adversarial ordinal classification predictor with weighted squared loss is obtained by choosing the parameter θ that minimizes the empirical risk of the surrogate loss function:*

$$AL^{sq\text{-}w}(\mathbf{f}, y, \alpha) = \max \left\{ \max_{\substack{i,j,l \in [k] \\ i < l \leq j}} \frac{(2(j-l)+1)\left[f_i + \alpha(l-i)^2\right] + (2(l-i)-1)\left[f_j + \alpha(j-l)^2\right]}{2(j-i)},\ \max_i f_i \right\} - f_y. \tag{2.51}$$

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 $\text{AL}^{\text{0-1-w}}$, $\mathcal{O}(k)$ for $\text{AL}^{\text{ord-w}}$, and $\mathcal{O}(k^3)$ for $\text{AL}^{\text{sq-w}}$. The weight constant α does not change the runtime complexity.

2.4.5 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 α. The adversarial surrogate loss for classification with abstention is defined in Equation (2.14), 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 Equation (2.14) 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.

$$
\begin{array}{l}
\\
\\
\\
\text{1st block} \\
\\
\\
\hdashline
\\
\\
\\
\text{2nd block} \\
\\
\hdashline
\\
\text{3rd block} \\
\\
\end{array}
\begin{bmatrix}
0 & 1 & 1 & 1 & -1 \\
1 & 0 & 1 & 1 & -1 \\
1 & 1 & 0 & 1 & -1 \\
1 & 1 & 1 & 0 & -1 \\
\alpha & \alpha & \alpha & \alpha & -1 \\
\hdashline
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
\hdashline
1 & 1 & 1 & 1 & 0 \\
-1 & -1 & -1 & -1 & 0
\end{bmatrix}
\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ v \end{bmatrix}
\geq
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ -1 \end{bmatrix}. \tag{2.52}
$$

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 $\mathrm{AL}^{\mathrm{abstain}}$ as stated in Theorem 2.7 below.

Theorem 2.7. *An adversarial predictor for classification with abstention with the penalty for abstain option is α where $0 \leq \alpha \leq \frac{1}{2}$, is obtained by choosing the parameter θ that minimizes the empirical risk of the surrogate loss function:*

$$AL^{abstain}(\mathbf{f}, y, \alpha) = \max \left\{ \max_{i,j \in [k], i \neq j} (1-\alpha) f_i + \alpha f_j + \alpha,\ \max_i f_i \right\} - f_y. \tag{2.53}$$

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

$$D = \left\{ \begin{bmatrix} \mathbf{q} \\ v \end{bmatrix} = (1-\alpha) \begin{bmatrix} \mathbf{e}_i \\ 0 \end{bmatrix} + \alpha \begin{bmatrix} \mathbf{e}_j \\ 1 \end{bmatrix} \;\middle|\; \begin{matrix} i, j \in [k] \\ i \neq j \end{matrix} \right\} \cup \left\{ \begin{bmatrix} \mathbf{q} \\ v \end{bmatrix} = \begin{bmatrix} \mathbf{e}_i \\ 0 \end{bmatrix} \;\middle|\; i \in [k] \right\}. \tag{2.54}$$

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

Similar to the proof of Theorem 2.2, we next prove that $D \supseteq \operatorname{ext} \mathbb{C} = \{\mathbf{c} \in \mathbb{C} : \operatorname{rank}(\mathbf{A_c}) = k+1\}$, as $D \subseteq \mathbb{C}$ is again obvious. 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 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.

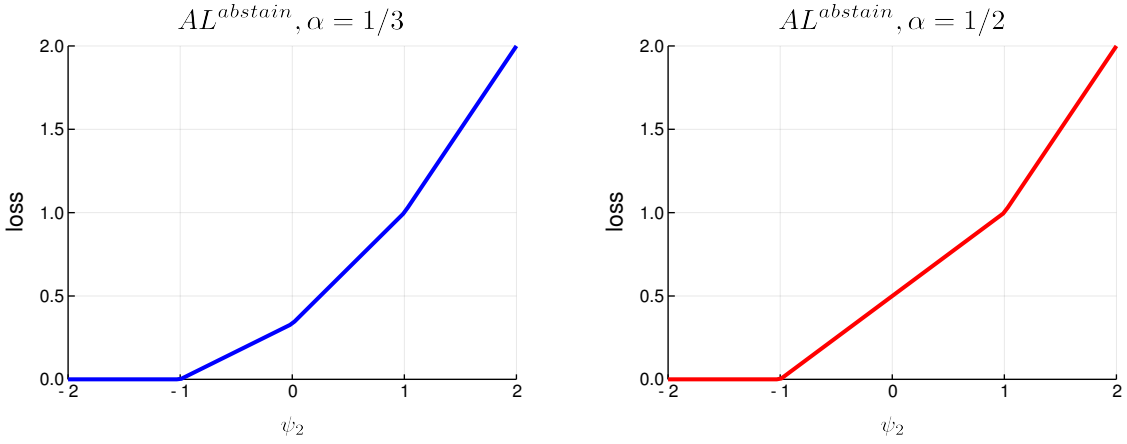
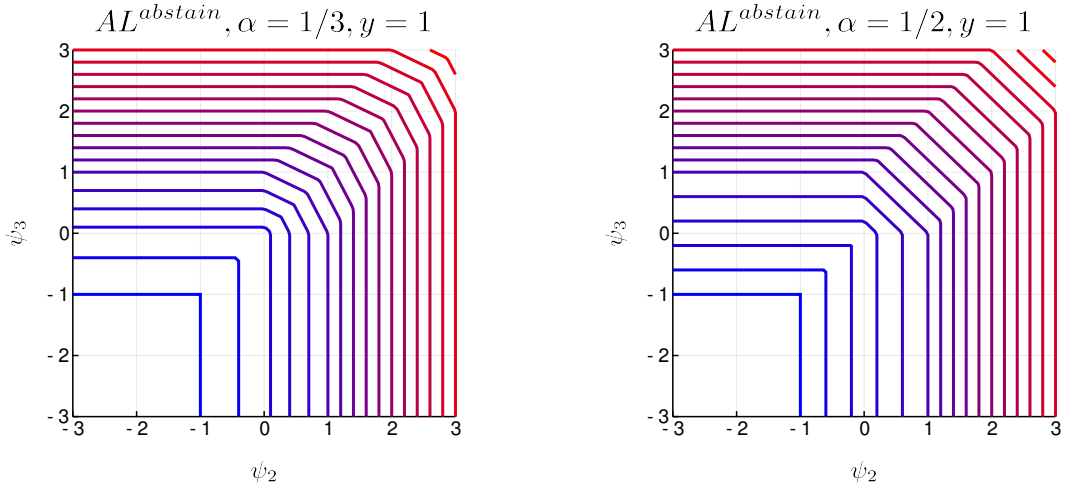


Figure 9. $\text{AL}^{\text{abstain}}$ evaluated over the space of potential differences ($\psi_{i,y} = f_i - f_y$; and $\psi_{i,i} = 0$) for binary prediction tasks when the true label is $y = 1$, where $\alpha = \frac{1}{3}$ (a), and $\alpha = \frac{1}{2}$ (b).



(a) (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).

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 largest and the second-largest 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 α 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.

2.4.6 General Multiclass Loss

For a general multiclass loss matrix $\mathbf{L}$, the extreme points of the polytope defined by the constraints in Equation (2.14) may not be easily characterized. Nevertheless, since the maxi-

mization in Equation (2.14) 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:

1. Simplex algorithm.

   The simplex algorithm (Dantzig, 1948; Dantzig, 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 Equation (2.14), $n = k + 1$.

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.

2.5 Prediction Formulation

The dual formulation of the adversarial prediction (Equation (2.6)) 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 $\theta^* = \text{argmin}_\theta \ \mathbb{E}_{\mathbf{X},Y\sim\tilde{P}} \ [AL(\mathbf{X}, Y, \theta)]$. In the prediction step, we use the optimal θ^* to make a label prediction given newly observed data. Although θ^* 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.

2.5.1 Probabilistic Prediction

Given a new data point $\mathbf{x}$ and its label y, and the optimal θ^*, we formulate the prediction minimax game based on Equation (2.6) by flipping the optimization order between the predictor and the adversary player:

$$\min_{\hat{P}(\hat{Y}|\mathbf{x})} \max_{\check{P}(\check{Y}|\mathbf{x})} \mathbb{E}_{\hat{Y}|\mathbf{x}\sim\hat{P};\check{Y}|\mathbf{x}\sim\check{P}} \left[\text{loss}(\hat{Y}, \check{Y}) + \theta^{*\intercal} \left(\phi(\mathbf{x}, \check{Y}) - \phi(\mathbf{x}, y)\right)\right]. \tag{2.55}$$

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:

$$\min_{\mathbf{p}\in\Delta} \max_{\mathbf{q}\in\Delta} \mathbf{p}^\intercal \mathbf{L}\mathbf{q} + \mathbf{f}^\intercal \mathbf{q} - f_y. \tag{2.56}$$

Even though the ground truth label y serves an important role in the learning step (Equation (2.6)), 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 Equation (2.56) can be removed, resulting in the following probabilistic prediction formulation:

$$\mathbf{p}^* = \operatorname*{argmin}_{\mathbf{p}\in\Delta} \max_{\mathbf{q}\in\Delta} \mathbf{p}^\intercal \mathbf{L}\mathbf{q} + \mathbf{f}^\intercal \mathbf{q}. \tag{2.57}$$

2.5.2 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 θ^*, the predicted label is computed by choosing the label that maximizes the potential value, i.e.,

   $$y^* = \operatorname*{argmax}_i f_i, \qquad \text{where: } f_i = \theta^{*\intercal}\phi(\mathbf{x}, i). \tag{2.58}$$

   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 Equation (2.57). Then the algorithm chooses the label that maximizes the conditional probability, i.e.,

$$y^* = \operatorname*{argmax}_i p_i^*, \qquad \text{where: } \mathbf{p}^* = \operatorname*{argmin}_{\mathbf{p}\in\Delta} \max_{\mathbf{q}\in\Delta} \mathbf{p}^\intercal \mathbf{L}\mathbf{q} + \mathbf{f}^\intercal \mathbf{q}. \tag{2.59}$$

   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).

2.5.3 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 (Equation (2.57)) without the need to use

general purpose LP solver. The minimax game in Equation (2.57) can be equivalently written in the standard LP form as:

$$\begin{aligned} &\min_{\mathbf{p},v} v \\ &\text{s.t.: } v \geq \mathbf{L}_{(:,i)}{}^{\intercal}\mathbf{p} + f_i, \quad \forall i \in [k] \\ &\qquad \mathbf{p} \in \mathbb{R}_{+}^{k+1}, \\ &\qquad \mathbf{p}^{\intercal}\mathbf{1} = 1, \end{aligned} \tag{2.60}$$

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.,

$$\mathbf{L} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ \alpha & \alpha & \alpha & \alpha \end{bmatrix} \tag{2.61}$$

in a four-class classification, where α is the penalty for abstaining (c.f. Section 2.4.5). 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 2.8. *Let α be the penalty for abstaining where $0 \leq \alpha \leq \frac{1}{2}$, θ^* 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*

canonical vector. Then the predictor's optimal probability $\mathbf{p}^*$ *of Equation* (2.60) *for the task of classification with abstention can be directly computed as:*

$$\mathbf{p}^* = \begin{bmatrix} \mathbf{e}_{i^*} \\ 0 \end{bmatrix} \text{ if } f_{i^*} - f_{j^*} \geq 1 \qquad \text{and} \qquad \mathbf{p}^* = \begin{bmatrix} (f_{i^*} - f_{j^*})\mathbf{e}_{i^*} \\ 1 - f_{i^*} + f_{j^*} \end{bmatrix} \text{ if } f_{i^*} - f_{j^*} < 1. \tag{2.62}$$

Proof. Based on Theorem 2.7, the optimal objective value of Equation (2.60) is exactly the value of $\mathrm{AL}^{\mathrm{abstain}}(\mathbf{f}, y, \alpha) + f_y$, 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 Equation (2.60) 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 2.4. *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:*

$$y^* = \begin{cases} i^* & f_{i^*} - f_{j^*} \geq \frac{1}{2} \\ abstain & otherwise \end{cases} \tag{2.63}$$

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 Equation (2.57), which costs $\mathcal{O}(k^{3.5})$.

2.6 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 δ is said

to be Fisher consistent with respect to the loss ℓ if the minimizer f^* of the surrogate loss reaches the Bayes optimal risk, i.e.:

$$f^* \in \operatorname*{argmin}_f \mathbb{E}_{Y|\mathbf{x}\sim P}\left[\delta_f(\mathbf{x}, Y)\right] \;\Rightarrow\; \mathbb{E}_{Y|\mathbf{x}\sim P}\left[\ell_{f^*}(\mathbf{x}, Y)\right] = \min_f \mathbb{E}_{Y|\mathbf{x}\sim P}\left[\ell_f(\mathbf{x}, Y)\right]. \tag{2.64}$$

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 ℓ_f has a similar meaning.

2.6.1 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 Equation (2.64) is equivalent to:

$$\mathbb{E}_{Y|\mathbf{x}\sim P}\left[\ell(\operatorname*{argmax}_{y'} f^*(\mathbf{x}, y'), Y)\right] = \min_f \mathbb{E}_{Y|\mathbf{x}\sim P}\left[\ell(\operatorname*{argmax}_{y'} f(\mathbf{x}, y'), Y)\right]. \tag{2.65}$$

Since f is optimized over all measurable functions, the condition in Equation (2.64) can be further simplified as

$$\begin{aligned} & f^* \in \operatorname*{argmin}_f \mathbb{E}_{Y|\mathbf{x}\sim P}\left[\delta_f(\mathbf{x}, Y)\right] \\ \Rightarrow\; & \operatorname*{argmax}_{y'} f^*(\mathbf{x}, y') \subseteq \operatorname*{argmin}_{y'} \mathbb{E}_{Y|\mathbf{x}\sim P}\left[\ell(y', Y)\right], \qquad \forall \mathbf{x} \in \mathcal{X}. \end{aligned} \tag{2.66}$$

Using the potential scoring function notation $f(\mathbf{x}, y)$, the adversarial surrogate loss in Equation (2.12) can be equivalently written as:

$$AL_f(\mathbf{x}, y) = \max_{\check{P}(\check{Y}|\mathbf{x})} \min_{\hat{P}(\hat{Y}|\mathbf{x})} \mathbb{E}_{\hat{Y}|\mathbf{x}\sim\hat{P};\check{Y}|\mathbf{x}\sim\check{P}} \left[\text{loss}(\hat{Y}, \check{Y}) + f(\mathbf{x}, \check{Y}) - f(\mathbf{x}, y)\right]. \tag{2.67}$$

Then, the Fisher consistency condition for the adversarial surrogate loss AL_f becomes:

$$\begin{aligned} f^* \in \mathcal{F}^* &\triangleq \operatorname*{argmin}_f \mathbb{E}_{Y|\mathbf{x}\sim P} \left[\text{AL}_f(\mathbf{x}, Y)\right] \\ \Rightarrow \operatorname*{argmax}_y f^*(\mathbf{x}, y) &\subseteq \mathcal{Y}^\diamond \triangleq \operatorname*{argmin}_{y'} \mathbb{E}_{Y|\mathbf{x}\sim P}[\text{loss}(y', Y)]. \end{aligned} \tag{2.68}$$

In the sequel, we will show that the condition in Equation (2.68) 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 2.9 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 2.10.

Theorem 2.9. *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 $y' \neq y$. 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 distribu-

tion $P(Y|\mathbf{x})$. So, all $\mathbf{p}$, $\mathbf{q}$, and $\mathbf{d}$ lie in the k dimensional probability simplex Δ, where k is the number of classes. Let $\mathbf{L}$ be a k-by-k loss matrix whose (y, y')-th entry is $\text{loss}(y, y')$. Let $\mathbf{f} \in \mathbb{R}^k$ be the vector encoding of the value of f at all classes. The definition of f^* in Equation (2.68) now becomes:

$$\mathbf{f}^* \in \operatorname*{argmin}_{\mathbf{f}} \max_{\mathbf{q}\in\Delta} \min_{\mathbf{p}\in\Delta} \{\mathbf{f}^\intercal \mathbf{q} + \mathbf{p}^\intercal \mathbf{L}\mathbf{q} - \mathbf{d}^\intercal \mathbf{f}\} = \operatorname*{argmin}_{\mathbf{f}} \max_{\mathbf{q}\in\Delta} \left\{ \mathbf{f}^\intercal \mathbf{q} + \min_y (\mathbf{L}\mathbf{q})_y - \mathbf{d}^\intercal \mathbf{f} \right\}. \tag{2.69}$$

Since $\mathcal{Y}^\diamond \triangleq \operatorname{argmin}_y \mathbb{E}_{Y|\mathbf{x}\sim P}[\text{loss}(y, Y)]$ (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 Equation (2.69) under $\mathbf{f}^*$. As a result:

$$\mathbf{d} \in \operatorname*{argmax}_{\mathbf{q}\in\Delta} \left\{ \mathbf{q}^\intercal \mathbf{f}^* + \min_y (\mathbf{L}\mathbf{q})_y \right\}. \tag{2.70}$$

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

$$\left(\mathbf{f}^* + \mathbf{L}_{(y^\diamond,:)}{}^\intercal\right)^\intercal (\mathbf{u} - \mathbf{d}) \leq 0 \quad \forall \mathbf{u} \in \Delta, \tag{2.71}$$

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 Equation (2.70) 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., $f^*_y + \text{loss}(y^\diamond, y)$ 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 2.9 to the case where the Bayes optimal prediction may have ties.

Theorem 2.10. *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 $\operatorname{argmin}_{y'} \sum_{y \in \mathcal{Y}^\diamond} \alpha_y loss(y, y') \subseteq \mathcal{Y}^\diamond$ *for all* $\alpha_{(\cdot)} \geq 0$ *and* $\sum_{y \in \mathcal{Y}^\diamond} \alpha_y = 1$*, then* $\operatorname{argmax}_y f^*(\mathbf{x}, y) \subseteq \mathcal{Y}^\diamond$ *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 2.9. Let $\mathcal{Y}^\diamond \triangleq \operatorname{argmin}_y(\mathbf{Ld})_y$ which is not necessarily a singleton. The analysis in the proof of Theorem 2.9 carries over to this case, except for Equation (2.71). Denote $h(\mathbf{q}) \triangleq \mathbf{q}^\intercal \mathbf{f}^* + \min_y(\mathbf{Lq})_y$. The subdifferential of $-h(\mathbf{q})$ evaluated at $\mathbf{q} = \mathbf{d}$ is the set:

$$\partial(-h)(\mathbf{d}) = \{-\mathbf{f}^* - \mathbf{v} \mid \mathbf{v} \in \mathbf{conv}\{\mathbf{L}_{(y^\diamond,:)}{}^\intercal \mid y^\diamond \in \mathcal{Y}^\diamond\}\}, \tag{2.72}$$

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:

$$\mathbf{g}^\intercal(\mathbf{u} - \mathbf{d}) \geq 0 \quad \forall \mathbf{u} \in \Delta. \tag{2.73}$$

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 Equation (2.72), $-\mathbf{g} - \mathbf{f}^*$ 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. Equation (2.68)):

$$\mathcal{F}^*_{\mathbf{x}} = \left\{ \mathbf{f}^* = c\mathbf{1} - \sum_{y \in \mathcal{Y}^\diamond} \alpha_y \mathbf{L}_{(y,:)}{}^\intercal \;\middle|\; \alpha_{(\cdot)} \geq 0,\ \sum_{y \in \mathcal{Y}^\diamond} \alpha_y = 1,\ c \in \mathbb{R} \right\}. \tag{2.74}$$

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 Equation (2.74) 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 $f^* \in \mathcal{F}^*_{\mathbf{x}}$. □

2.6.2 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 2.5.3), 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 2.11. *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 Δ^k, while the predictor's probability mass $\mathbf{p}$ lies in the l dimensional probability simplex Δ^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:

$$\mathbf{f}^* \in \operatorname*{argmin}_{\mathbf{f}} \max_{\mathbf{q}\in\Delta^k} \min_{\mathbf{p}\in\Delta^l} \left\{\mathbf{f}^\intercal\mathbf{q} + \mathbf{p}^\intercal\mathbf{L}\mathbf{q} - \mathbf{d}^\intercal\mathbf{f}\right\}. \tag{2.75}$$

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 Equation (2.75) under $\mathbf{f}^*$.

Here we use the probabilistic prediction scheme as mentioned in Equation (2.57). The consistency condition in Equation (2.64) 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.,

$$\mathbf{p}^{\diamond\intercal}\mathbf{L}\mathbf{d} = \min_y(\mathbf{L}\mathbf{d})_y, \quad \text{where} \quad \mathbf{p}^\diamond = \operatorname*{argmin}_{\mathbf{p}\in\Delta^l} \max_{\mathbf{q}\in\Delta^k} \mathbf{p}^\intercal\mathbf{L}\mathbf{q} + \mathbf{f}^{*\intercal}\mathbf{q}. \tag{2.76}$$

Since the maximization over $\mathbf{q}$ in Equation (2.75) does not depend on $\mathbf{d}^\intercal\mathbf{f}$, we know that $\mathbf{d}$ is also an optimal solution of $\operatorname{argmax}_{\mathbf{q}\in\Delta^k} \min_{\mathbf{p}\in\Delta^l} \mathbf{p}^\intercal\mathbf{L}\mathbf{q} + \mathbf{f}^{*\intercal}\mathbf{q}$. Then, based on the minimax duality theorem (Von Neumann and Morgenstern, 1945), we know that:

$$\mathbf{p}^{\diamond\intercal}\mathbf{L}\mathbf{d} + \mathbf{f}^{*\intercal}\mathbf{d} = \min_{\mathbf{p}\in\Delta^l} \mathbf{p}^\intercal\mathbf{L}\mathbf{d} + \mathbf{f}^{*\intercal}\mathbf{d}. \tag{2.77}$$

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. □

2.7 Optimization

The goal of a learning algorithm in the adversarial prediction framework is to obtain the optimal Lagrange dual variable θ that enforces the adversary's probability distribution to reside within the moment matching constraints in Equation (2.5). In the risk minimization perspective (Equation (2.6)), it is equivalent to finding the parameter θ that minimizes the adversarial surrogate loss (AL) in Equation (2.12). To find the optimal θ, we employ (sub)-gradient methods to optimize our convex objective.

2.7.1 Subgradient-Based Convex Optimization

The risk minimization perspective of adversarial prediction framework (Equation (2.6)) can be written as:

$$\min_{\theta} \mathbb{E}_{\mathbf{X},Y\sim\tilde{P}} \left[AL(\mathbf{X}, Y, \theta)\right] \tag{2.78}$$
$$\text{where: } AL(\mathbf{x}, y, \theta) = \max_{\mathbf{q}\in\Delta} \min_{\mathbf{p}\in\Delta} \mathbf{p}^\intercal \mathbf{L}\mathbf{q} + \theta^\intercal \left[\sum_j q_j \phi(\mathbf{x}, j) - \phi(\mathbf{x}, y)\right].$$

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

$$\partial_\theta \, \mathbb{E}_{\mathbf{X},Y\sim\tilde{P}} \left[AL(\mathbf{X}, Y, \theta)\right] = \mathbb{E}_{\mathbf{X},Y\sim\tilde{P}} \left[\partial_\theta \, AL(\mathbf{X}, Y, \theta)\right]. \tag{2.79}$$

Theorem 2.12 describes the subgradient of the adversarial surrogate loss with respect to θ.

Theorem 2.12. *Given θ, suppose the set of optimal $\mathbf{q}$ for the maximin inside the AL is Q^*:*

$$Q^* = \operatorname*{argmax}_{\mathbf{q}\in\Delta} \min_{\mathbf{p}\in\Delta} \left\{ \mathbf{p}^\intercal \mathbf{L}\mathbf{q} + \theta^\intercal \left[\textstyle\sum_j q_j \phi(\mathbf{x}, j) - \phi(\mathbf{x}, y) \right] \right\}. \tag{2.80}$$

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

$$\partial_\theta \; AL(\mathbf{x}, y, \theta) = \mathbf{conv} \left\{ \textstyle\sum_j q_j^* \phi(\mathbf{x}, j) - \phi(\mathbf{x}, y) \;\middle|\; \mathbf{q} \in Q^* \right\}. \tag{2.81}$$

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 θ. 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 θ, and φ 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 2.5. *The subdifferential of $AL^{0\text{-}1}(\mathbf{x}, y, \theta)$ with respect to θ includes:*

$$\partial_\theta \; AL^{0\text{-}1}(\mathbf{x}, y, \theta) \ni \tfrac{1}{|S^*|} \textstyle\sum_{j\in S^*} \phi(\mathbf{x}, j) - \phi(\mathbf{x}, y), \tag{2.82}$$

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

$$S^* \in \underset{S\subseteq[k],\, S\neq\emptyset}{\operatorname{argmax}} \frac{\sum_{j\in S} \theta^\intercal \phi(\mathbf{x}, j) + |S| - 1}{|S|}. \tag{2.83}$$

Corollary 2.6. *The subdifferential of $AL^{ord}(\mathbf{x}, y, \theta)$ with respect to θ includes:*

$$\partial_\theta\ AL^{ord}(\mathbf{x}, y, \theta) \ni \tfrac{1}{2}\left(\phi(\mathbf{x}, i^*) + \phi(\mathbf{x}, j^*)\right) - \phi(\mathbf{x}, y), \tag{2.84}$$

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

$$(i^*, j^*) \in \underset{i,j\in[k]}{\operatorname{argmax}} \frac{\theta^\intercal \phi(\mathbf{x}, i) + \theta^\intercal \phi(\mathbf{x}, j) + j - i}{2}. \tag{2.85}$$

Corollary 2.7. *The subdifferential of $AL^{abstain}(\mathbf{x}, y, \theta, \alpha)$ where $0 \leq \alpha \leq \frac{1}{2}$ with respect to θ includes:*

$$\partial_\theta\ AL^{abstain}(\mathbf{x}, y, \theta, \alpha) \ni \begin{cases} (1-\alpha)\phi(\mathbf{x}, i^*) + \alpha\phi(\mathbf{x}, j^*) - \phi(\mathbf{x}, y) & g(\mathbf{x}, y, \theta, \alpha) > h(\mathbf{x}, y, \theta, \alpha) \\ \phi(\mathbf{x}, l^*) - \phi(\mathbf{x}, y) & \textit{otherwise}, \end{cases} \tag{2.86}$$

where:

$$g(\mathbf{x}, y, \theta, \alpha) = \max_{i,j \in [k], i \neq j} (1-\alpha)\, f_i + \alpha f_j + \alpha, \quad h(\mathbf{x}, y, \theta, \alpha) = \max_l f_l, \tag{2.87}$$

$$(i^*, j^*) \in \operatorname*{argmax}_{i,j \in [k], i \neq j} (1-\alpha)\, f_i + \alpha f_j + \alpha, \quad l^* = \operatorname*{argmax}_l f_l, \tag{2.88}$$

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 AL^{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 θ 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.

2.7.2 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,

$$\min_{\theta} \mathbb{E}_{\mathbf{X},Y\sim\tilde{P}} \frac{\lambda}{2}\|\theta\|^2 + AL(\mathbf{X}, Y, \theta), \tag{2.89}$$

where λ 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:

$$\frac{\lambda}{2}\|\theta\|^2 + AL(\mathbf{x}_{i_t}, y_{i_t}, \theta), \tag{2.90}$$

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

$$\begin{aligned} \partial_\theta^{(t)} =& \lambda\theta^{(t)} + \textstyle\sum_j q_j^{*(t)}\phi(\mathbf{x}_{i_t}, j) - \phi(\mathbf{x}_{i_t}, y_{i_t}), \\ \text{where: } \mathbf{q}^{*(t)} =& \operatorname*{argmax}_{\mathbf{q}\in\Delta} \min_{\mathbf{p}\in\Delta} \mathbf{p}^\intercal \mathbf{L}\mathbf{q} + \mathbf{f}^{(t)\intercal}\mathbf{q} - f_{y_{i_t}}^{(t)}, \\ f_j^{(t)} =& \theta^{(t)\intercal}\phi(\mathbf{x}_{i_t}, j). \end{aligned} \tag{2.91}$$

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,

$$\theta^{(t+1)} = \theta^{(t)} - \eta^{(t)}\partial_\theta^{(t)} = \theta^{(t)} - \tfrac{1}{\lambda t}\partial_\theta^{(t)}. \tag{2.92}$$

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 Equation (2.91), then the update steps can be written as:

$$\theta^{(t+1)} = (1 - \tfrac{1}{t})\theta^{(t)} - \tfrac{1}{\lambda t}\mathbf{g}^{(t)}. \tag{2.93}$$

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

$$\theta^{(t+1)} = -\frac{1}{\lambda t}\sum_{l=1}^{t}\mathbf{g}^{(l)}, \tag{2.94}$$

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

$$\theta^{(t+1)} = -\frac{1}{\lambda t}\sum_{l=1}^{t}\sum_{j=1}^{k} q_j^{*(l)}\phi(\mathbf{x}_{i_l}, j) - \phi(\mathbf{x}_{i_l}, y_{i_l}), \tag{2.95}$$

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

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

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 Equation (2.95) can be equivalently written as:

$$\theta^{(t+1)} = -\frac{1}{\lambda t}\sum_{i=1}^{n}\sum_{j=1}^{k} \alpha_{(i,j)}^{(t+1)}\phi(\mathbf{x}_i, j), \tag{2.96}$$

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:

$$f_j^{(t)} = \theta^{(t)\intercal}\phi(\mathbf{x}_{i_t}, j) = -\frac{1}{\lambda t}\sum_{i'}^{n}\sum_{j'}^{k} \alpha_{(i',j')}^{(t)}\ \phi(\mathbf{x}_{i'}, j')\cdot\phi(\mathbf{x}_{i_t}, j). \tag{2.97}$$

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,

$$f_j^{(t)} = \theta^{(t)\intercal}\omega(\phi(\mathbf{x}_{i_t}, j)) = -\frac{1}{\lambda t}\sum_{i'}^{n}\sum_{j'}^{k}\alpha_{(i',j')}^{(t)}\omega(\phi(\mathbf{x}_{i'}, j')) \cdot \omega(\phi(\mathbf{x}_{i_t}, j)) \tag{2.98}$$

$$= -\frac{1}{\lambda t}\sum_{i'}^{n}\sum_{j'}^{k}\alpha_{(i',j')}^{(t)}K(\phi(\mathbf{x}_{i'}, j'), \phi(\mathbf{x}_{i_t}, j)). \tag{2.99}$$

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

Algorithm 1 PEGASOS algorithm for adversarial surrogate losses with kernel trick

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

2.8 Experiments

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

2.8.1 Experiments for Multiclass Zero-One Loss Metric

We evaluate the performance of the $AL^{0\text{-}1}$ classifier and compare it with the three most popular multiclass SVM formulations: the WW (Weston et al., 1999), the CS (Crammer and Singer, 2002), and the 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 I). 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.

TABLE I. Properties of the datasets for the zero-one loss metric experiments.

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

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 γ 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 γ 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 γ from $2^i \gamma_0, i = \{-2, -1, 0, 1, 2\}$ in the second stage, where C_0 and γ_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.

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 II. 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 datasets 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 $\text{AL}^{0\text{-}1}$ classifier performs competitively with the WW and CS models

TABLE II. 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 $\alpha = 0.05$).

D	Linear Kernel				Gaussian Kernel			
	$AL^{0\text{-}1}$	WW	CS	LLW	$AL^{0\text{-}1}$	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

with 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 $AL^{0\text{-}1}$ 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 minimization) multiclass classifier that enjoys both strong theoretical properties (Fisher consistency) and empirical performance.

2.8.2 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 III.

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 (RED^{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 (RED^{mc}) (Li and Lin, 2007), cost-sensitive one-sided support vector regression (CSOSR)

TABLE III. Properties of the datasets for the ordinal classification experiments.

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

(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 λ. In the first stage, the possible values for λ are $2^{-i}, i = \{1, 3, 5, 7, 9, 11, 13\}$. Using the best λ in the first stage, we set the possible values for λ in the second stage as $2^{\frac{i}{2}}\lambda_0, i = \{-3, -2, -1, 0, 1, 2, 3\}$, where λ_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 IV and Table V. 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.

As we can see from Table IV, 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, $\text{AL}^{\text{ord-th}}$, RED^{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 RED^{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-

TABLE IV. 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 $\alpha = 0.05$).

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

based models, $\text{AL}^{\text{ord-mc}}$ and RED^{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 RED^{mc}. A notable exception is the CSOVO model which perform very well on the `stocks`, and `boston` datasets.

TABLE V. 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 $\alpha = 0.05$).

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

In the Gaussian kernel experiment, we can see from Table V 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.

2.8.3 Experiments for Multiclass Classification with Abstention

We conduct experiments for classification with abstention tasks using the same dataset as in the multiclass zero-one experiments (Table I). 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:

$$\text{loss}(\hat{y}, y) = \begin{cases} \alpha & \hat{y} = k+1 \\ I(\hat{y} \neq y) & \text{otherwise,} \end{cases} \tag{2.100}$$

where $\hat{y} = k+1$ indicates an abstain prediction, and α 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 λ) and the kernel parameter γ 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 2.4. 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 τ 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 VI. 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 VI 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,

TABLE VI. 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 $\alpha = 0.05$).

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

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`).

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 VI. 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.

2.9 Conclusions and Future Works

In this section, 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 multi-

class classification by simultaneously: guaranteeing Fisher consistency, enabling computational efficiency 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.

CHAPTER 3

PERFORMANCE-ALIGNED ADVERSARIAL GRAPHICAL MODELS

(This chapter was previously published as "Distributionally Robust Graphical Models" (Fathony et al., 2018b) in the Advances in Neural Information Processing Systems 31 (NeurIPS 2018).)

3.1 Introduction

Learning algorithms must consider complex relationships between variables to provide useful predictions in many structured prediction problems. These complex relationships are often represented using graphs to convey the independence assumptions being employed. For example, chain structures are used when modeling sequences like words and sentences (Manning and Schütze, 1999), tree structures are popular for natural language processing tasks that involve prediction for entities in parse trees (Cohn and Blunsom, 2005; Hatori et al., 2008; Sadeghian et al., 2016), and lattice structures are often used for modeling images (Nowozin et al., 2011). The most prevalent methods for learning with graphical structure are probabilistic graphical models (e.g., conditional random fields (CRFs) (Lafferty et al., 2001)) and large margin models (e.g., structured support vector machines (SSVMs) (Tsochantaridis et al., 2005) and maximum margin Markov networks (M^3Ns) (Taskar et al., 2005a)). Both types of models have unique advantages and disadvantages. CRFs with sufficiently expressive feature representation are consistent estimators of the marginal probabilities of variables in cliques of the graph (Li, 2009),

but are oblivious to the evaluative loss metric during training. On the other hand, SSVMs directly incorporate the evaluative loss metric in the training optimization, but lack consistency guarantees for multiclass settings (Tewari and Bartlett, 2007; Liu, 2007).

To address these limitations, we propose adversarial graphical models (AGM), a distributionally robust framework for leveraging graphical structure among variables that provides both the flexibility to incorporate customized loss metrics during training as well as the statistical guarantee of Fisher consistency for a chosen loss metric. Our approach is based on a robust adversarial formulation (Topsøe, 1979; Grünwald and Dawid, 2004; Asif et al., 2015) that seeks a predictor that minimizes a loss metric in the worst-case given the statistical summaries of the empirical distribution. 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 match the statistical summaries of empirical training data via moment matching constraints, as defined by a graphical structure.

Our AGM framework accepts a variety of loss metrics. A notable example that connects our framework to previous models is the logarithmic loss metric. The conditional random field (CRF) model (Lafferty et al., 2001) can be viewed as the robust predictor that best minimizes the logarithmic loss metric in the worst-case subject to moment matching constraints. We focus on a family of loss matrices that additively decomposes over each variable and is defined only based on the label values of the predictor and evaluator. For examples, the additive zero-one (the Hamming loss), ordinal regression (absolute), and cost sensitive metrics fall into this family of loss metrics. We propose efficient exact algorithms for learning and prediction for

graphical structures with low treewidth. Finally, we experimentally demonstrate the benefits of our framework compared with the previous models on structured prediction tasks.

3.2 Background and related works

3.2.1 Structured prediction, Fisher consistency, and graphical models

The structured prediction task is to simultaneously predict correlated label variables $\mathbf{y} \in \boldsymbol{\mathcal{Y}}$—often given input variables $\mathbf{x} \in \boldsymbol{\mathcal{X}}$—to minimize a loss metric (e.g., loss : $\boldsymbol{\mathcal{Y}} \times \boldsymbol{\mathcal{Y}} \rightarrow \mathbb{R}$) with respect to the true label values $\tilde{\mathbf{y}}$. This is in contrast with classification methods that predict one single variable y. Given a distribution over the multivariate labels, $P(\mathbf{y})$, **Fisher consistency** is a desirable characteristic that requires a learning method to produce predictions $\hat{\mathbf{y}}$ that minimize the expected loss of this distribution, $\hat{\mathbf{y}}^* \in \operatorname{argmin}_{\hat{\mathbf{y}}} \mathbb{E}_{\mathbf{Y} \sim \tilde{P}}[\text{loss}(\hat{\mathbf{y}}, \mathbf{Y})]$, under ideal learning conditions (i.e., trained from the true data distribution using a fully expressive feature representation).

To reduce the complexity of the mappings from $\boldsymbol{\mathcal{X}}$ to $\boldsymbol{\mathcal{Y}}$ being learned, independence assumptions and more restrictive representations are employed. In probabilistic graphical models, such as Bayesian networks (Pearl, 1985) and random fields (Lafferty et al., 2001), these assumptions are represented using a graph over the variables. For graphs with arbitrary structure, inference (i.e., computing posterior probabilities or maximal value assignments) requires exponential time in terms of the number of variables (Cooper, 1990). However, this run-time complexity reduces to be polynomial in terms of the number of predicted variables for graphs with low treewidth (e.g., chains, trees, cycles).

3.2.2 Conditional random fields as robust multivariate log loss minimization

Following ideas from robust Bayes decision theory (Topsøe, 1979; Grünwald and Dawid, 2004) and distributional robustness (Delage and Ye, 2010), the conditional random field (Lafferty et al., 2001) can be derived as a robust minimizer of the logarithmic loss subject to moment-matching constraints:

$$\min_{\hat{P}(\cdot|\mathbf{x})} \max_{\check{P}(\cdot|\mathbf{x})} \mathbb{E}_{\substack{\mathbf{X}\sim\tilde{\mathbf{P}};\\ \check{\mathbf{Y}}|\mathbf{X}\sim\check{P}}} \left[-\log \hat{P}(\check{\mathbf{Y}}|\mathbf{X})\right] \text{ such that: } \mathbb{E}_{\substack{\mathbf{X}\sim\tilde{\mathbf{P}};\\ \check{\mathbf{Y}}|\mathbf{X}\sim\check{P}}} \left[\Phi(\mathbf{X},\check{\mathbf{Y}})\right] = \mathbb{E}_{\mathbf{X},\mathbf{Y}\sim\tilde{P}} \left[\Phi(\mathbf{X},\mathbf{Y})\right], \tag{3.1}$$

where $\Phi : \boldsymbol{\mathcal{X}} \times \boldsymbol{\mathcal{Y}} \to \mathbb{R}^k$ are feature functions that typically decompose additively over subsets of variables. Under this perspective, the predictor $\hat{P}$ seeks the conditional distribution that minimizes log loss against an adversary $\check{P}$ seeking to choose an evaluation distribution that approximates training data statistics, while otherwise maximizing log loss. As a result, the predictor is robust not only to the training sample $\tilde{P}$, but all distributions with matching moment statistics (Grünwald and Dawid, 2004).

The saddle point for Equation (3.1) is obtained by the parametric conditional distribution $\hat{P}_\theta(\mathbf{y}|\mathbf{x}) = \check{P}_\theta(\mathbf{y}|\mathbf{x}) = e^{\theta\cdot\Phi(\mathbf{x},\mathbf{y})} / \sum_{\mathbf{y}'\in\boldsymbol{\mathcal{Y}}} e^{\theta\cdot\Phi(\mathbf{x},\mathbf{y}')}$ with parameters θ chosen by maximizing the data likelihood: $\operatorname{argmax}_\theta \mathbb{E}_{\mathbf{X},\mathbf{Y}\sim\tilde{\mathbf{P}}} \left[\log \hat{P}_\theta(\mathbf{Y}|\mathbf{X})\right]$. The decomposition of the feature function into additive clique features, $\Phi_i(\mathbf{x},\mathbf{y}) = \sum_{c\in\mathcal{C}_i} \phi_{c,i}(\mathbf{x}_c,\mathbf{y}_c)$, can be represented graphically by connecting the variables within cliques with undirected edges. Dynamic programming algorithms (e.g., junction tree) allow the exact likelihood to be computed in run time that is exponential in terms of the treewidth of the resulting graph (Cowell et al., 2006).

Predictions for a particular loss metric are then made using the Bayes optimal prediction for the estimated distribution: $\mathbf{y}^* = \text{argmin}_{\mathbf{y}} \mathbb{E}_{\hat{\mathbf{Y}}|\mathbf{x}\sim\hat{P}_\theta}[\text{loss}(\mathbf{y}, \hat{\mathbf{Y}})]$. This two-stage prediction approach can create inefficiencies when learning from limited amounts of data since optimization may focus on accurately estimating probabilities in portions of the input space that have no impact on the decision boundaries of the Bayes optimal prediction. Rather than separating the prediction task from the learning process, we incorporate the evaluation loss metric of interest into the robust minimization formulation of Equation (3.1) in this work.

3.2.3 Structured support vector machines

Structured support vector machines (SSVMs) (Joachims, 2005) and related maximum margin methods (Taskar et al., 2005a) directly incorporate the evaluation loss metric into the training process. This is accomplished by minimizing a hinge loss convex surrogate:

$$\text{hinge}_\theta(\tilde{\mathbf{y}}) = \max_{\mathbf{y}} \text{loss}(\mathbf{y}, \tilde{\mathbf{y}}) + \theta \cdot (\Phi(\mathbf{x}, \tilde{\mathbf{y}}) - \Phi(\mathbf{x}, \mathbf{y}))\,, \tag{3.2}$$

where θ represents the model parameters, $\tilde{\mathbf{y}}$ is the ground truth label, and $\Phi(\mathbf{x}, \mathbf{y})$ is a feature function that decomposes additively over subsets of variables.

Using a clique-based graphical representation of the potential function, and assuming the loss metric also additively decomposes into the same clique-based representation, SSVMs have a computational complexity similar to probabilistic graphical models. Specifically, finding the value assignment $\mathbf{y}$ that maximizes this loss-augmented potential can be accomplished using

dynamic programming in run time that is exponential in the graph treewidth (Cowell et al., 2006).

A key weakness of support vector machines in general is their lack of Fisher consistency; there are distributions for multiclass prediction tasks for which the SVM will not learn a Bayes optimal predictor, even when the models are given access to the true distribution and sufficiently expressive features, due to the disconnection between the Crammer-Singer hinge loss surrogate (Crammer and Singer, 2002) and the evaluation loss metric (i.e., the 0-1 loss in this case) (Liu, 2007). In practice, if the empirical data behaves similarly to those distributions (e.g., $P(\mathbf{y}|\mathbf{x})$ have no majority $\mathbf{y}$ for a specific input $\mathbf{x}$), the inconsistent model may perform poorly. This inconsistency extends to the structured prediction setting except in limited special cases (Zhang, 2004). We overcome these theoretical deficiencies in our approach by using an adversarial formulation that more closely aligns the training objective with the evaluation loss metric, while maintaining convexity.

3.2.4 Other related works

Distributionally robust learning. There has been a recent surge of interest in the machine learning community for developing distributionally robust learning algorithms. The proposed learning algorithms differ in the uncertainty sets used to provide robustness. Previous robust learning algorithms have been proposed under the F-divergence measures (which includes the popular KL-divergence and χ-divergence) (Namkoong and Duchi, 2016; Namkoong and Duchi, 2017; Hashimoto et al., 2018), the Wasserstein metric uncertainty set (Shafieezadeh-Abadeh et al., 2015; Esfahani and Kuhn, 2018; Chen and Paschalidis, 2018), and the moment

matching uncertainty set (Delage and Ye, 2010; Livni et al., 2012). Our robust adversarial learning approach differs from the previous approaches by focusing on the robustness in terms of the conditional distribution $P(\mathbf{y}|\mathbf{x})$ instead of the joint distribution $P(\mathbf{x}, \mathbf{y})$. Our approach seeks a predictor that is robust to the worst-case conditional label probability under the moment matching constraints. We do not impose any robustness to the training examples $\mathbf{x}$.

Consistent methods. A notable research interest in consistent methods for structured prediction tasks has also been observed. This line of works includes a consistent regularization approach that maps the original structured prediction problem into a kernel Hilbert space and employs a multivariate regression on the Hilbert space (Ciliberto et al., 2016), and a consistent quadratic surrogate for any structured prediction loss metric with a polynomial sample complexity analysis for the additive zero-one loss metric surrogate (Osokin et al., 2017). Our work differs from these lines of works in the focus on the structure. We focus on the graphical structures that model interaction between labels, whereas the previous works focus on the structure of the loss metric itself.

3.3 Adversarial Graphical Models

We propose adversarial graphical models (AGMs) to better align structured prediction with evaluation loss metrics in settings where the structured interaction between labels are represented in a graph.

3.3.1 Formulations

We construct a predictor that best minimizes a loss metric for the worst-case evaluation distribution that (approximately) matches the statistical summaries of empirical training data.

Our predictor is allowed to make a probabilistic prediction over all possible label assignments (denoted as $\hat{P}(\hat{\mathbf{y}}|\mathbf{x})$). However, instead of evaluating the prediction with empirical data (as commonly performed by empirical risk minimization formulations (Vapnik, 1998)), the predictor is pitted against an adversary that also makes a probabilistic prediction (denoted as $\check{P}(\check{\mathbf{y}}|\mathbf{x})$). The adversary is constrained to select its conditional distributions to match the statistical summaries of the empirical training distribution (denoted as $\tilde{P}$) via moment matching constraints on the feature functions Φ.

Definition 3.1. *The adversarial prediction method for structured prediction problems with graphical interaction between labels is:*

$$\min_{\hat{P}(\hat{\mathbf{y}}|\mathbf{x})} \max_{\check{P}(\check{\mathbf{y}}|\mathbf{x})} \mathbb{E}_{\substack{\mathbf{X}\sim\tilde{P};\\ \hat{\mathbf{Y}}|\mathbf{X}\sim\hat{P};\\ \check{\mathbf{Y}}|\mathbf{X}\sim\check{P}}} \Big[\mathit{loss}(\hat{\mathbf{Y}}, \check{\mathbf{Y}}) \Big] \text{ such that: } \mathbb{E}_{\substack{\mathbf{X}\sim\tilde{P};\\ \check{\mathbf{Y}}|\mathbf{X}\sim\check{P}}} \big[\Phi(\mathbf{X}, \check{\mathbf{Y}}) \big] = \tilde{\Phi}, \tag{3.3}$$

where the vector of feature moments, $\tilde{\Phi} = \mathbb{E}_{\mathbf{X},\mathbf{Y}\sim\tilde{P}}[\Phi(\mathbf{X}, \mathbf{Y})]$*, is measured from sample training data. The feature function* $\Phi(\mathbf{X}, \mathbf{Y})$ *contains features that are additively decomposed over cliques in the graph, e.g.* $\Phi(\mathbf{x}, \mathbf{y}) = \sum_c \phi(\mathbf{x}, \mathbf{y}_c)$.

We focus on pairwise graphical structures where the interactions between labels are defined over the edges (and nodes) of the graph. We also restrict the loss metric to a family of metrics that additively decompose over each y_i variable, i.e., $\text{loss}(\hat{\mathbf{y}}, \check{\mathbf{y}}) = \sum_{i=1}^{n} \text{loss}(\hat{y}_i, \check{y}_i)$. Directly solving the optimization in Equation (3.3) is impractical for reasonably-sized problems since $P(\mathbf{y}|\mathbf{x})$ grows exponentially with the number of predicted variables. Instead, we utilize the

method of Lagrange multipliers and the marginal formulation of the distributions of predictor and adversary to formulate a simpler dual optimization problem as stated in Theorem 3.1.

Theorem 3.1. *For the adversarial structured prediction with pairwise graphical structure and an additive loss metric, solving the optimization in Definition 1 is equivalent to solving the following expectation of maximin problems over the node and edge marginal distributions parameterized by Lagrange multipliers θ:*

$$\begin{aligned}\min_{\theta_e,\theta_v} \mathbb{E}_{\mathbf{X},\mathbf{Y}\sim\tilde{P}} \max_{\check{P}(\check{\mathbf{y}}|\mathbf{x})} \min_{\hat{P}(\hat{\mathbf{y}}|\mathbf{x})} \Big[& \textstyle\sum_i^n \sum_{\hat{y}_i,\check{y}_i} \hat{P}(\hat{y}_i|\mathbf{x})\check{P}(\check{y}_i|\mathbf{x}) loss(\hat{y}_i,\check{y}_i) \\ & + \textstyle\sum_{(i,j)\in E} \sum_{\check{y}_i,\check{y}_j} \check{P}(\check{y}_i,\check{y}_j|\mathbf{x}) \left[\theta_e \cdot \phi(\mathbf{x},\check{y}_i,\check{y}_j)\right] - \sum_{(i,j)\in E} \theta_e \cdot \phi(\mathbf{x},y_i,y_j) \\ & + \textstyle\sum_i^n \sum_{\check{y}_i} \check{P}(\check{y}_i|\mathbf{x}) \ \left[\theta_v \cdot \phi(\mathbf{x},\check{y}_i)\right] - \sum_i^n \theta_v \cdot \phi(\mathbf{x},y_i) \Big],\end{aligned} \tag{3.4}$$

where $\phi(\mathbf{x},y_i)$ is the node feature function for node i, $\phi(\mathbf{x},y_i,y_j)$ is the edge feature function for the edge connecting node i and j, E is the set of edges in the graphical structure, and θ_v and θ_e are the Lagrange dual variables for the moment matching constraints corresponding to the node and edge features, respectively. The optimization objective depends on the predictor's probability prediction $\hat{P}(\hat{\mathbf{y}}|\mathbf{x})$ only through its node marginal probabilities $\hat{P}(\hat{y}_i|\mathbf{x})$. Similarly, the objective depends on the adversary's probabilistic prediction $\check{P}(\check{\mathbf{y}}|\mathbf{x})$ only through its node and edge marginal probabilities, i.e., $\check{P}(\check{y}_i|\mathbf{x})$, and $\check{P}(\check{y}_i,\check{y}_j|\mathbf{x})$.

Proof.

$$\min_{\hat{P}(\hat{\mathbf{y}}|\mathbf{x})} \max_{\check{P}(\check{\mathbf{y}}|\mathbf{x})} \mathbb{E}_{\mathbf{X}\sim\tilde{P};\hat{\mathbf{Y}}|\mathbf{X}\sim\hat{P};\check{\mathbf{Y}}|\mathbf{X}\sim\check{P}} \Big[\text{loss}(\hat{\mathbf{Y}},\check{\mathbf{Y}})\Big] \tag{3.5}$$

$$\text{subject to: } \mathbb{E}_{\mathbf{X}\sim\tilde{P};\check{\mathbf{Y}}|\mathbf{X}\sim\check{P}}\left[\Phi(\mathbf{X},\check{\mathbf{Y}})\right] = \mathbb{E}_{\mathbf{X},\mathbf{Y}\sim\tilde{P}}\left[\Phi(\mathbf{X},\mathbf{Y})\right]$$

$$\overset{(a)}{=} \max_{\check{P}(\check{\mathbf{y}}|\mathbf{x})} \min_{\hat{P}(\hat{\mathbf{y}}|\mathbf{x})} \mathbb{E}_{\mathbf{X}\sim\tilde{P};\hat{\mathbf{Y}}|\mathbf{X}\sim\hat{P};\check{\mathbf{Y}}|\mathbf{X}\sim\check{P}} \Big[\text{loss}(\hat{\mathbf{Y}},\check{\mathbf{Y}})\Big] \tag{3.6}$$

$$\text{subject to: } \mathbb{E}_{\mathbf{X}\sim\tilde{P};\check{\mathbf{Y}}|\mathbf{X}\sim\check{P}}\left[\Phi(\mathbf{X},\check{\mathbf{Y}})\right] = \mathbb{E}_{\mathbf{X},\mathbf{Y}\sim\tilde{P}}\left[\Phi(\mathbf{X},\mathbf{Y})\right]$$

$$\overset{(b)}{=} \max_{\check{P}(\check{\mathbf{y}}|\mathbf{x})} \min_{\theta} \min_{\hat{P}(\hat{\mathbf{y}}|\mathbf{x})} \mathbb{E}_{\mathbf{X},\mathbf{Y}\sim\tilde{P};\hat{\mathbf{Y}}|\mathbf{X}\sim\hat{P};\check{\mathbf{Y}}|\mathbf{X}\sim\check{P}} \Big[\text{loss}(\hat{\mathbf{Y}},\check{\mathbf{Y}}) + \theta^{\mathrm{T}}\left(\Phi(\mathbf{X},\check{\mathbf{Y}}) - \Phi(\mathbf{X},\mathbf{Y})\right)\Big] \tag{3.7}$$

$$\overset{(c)}{=} \min_{\theta} \max_{\check{P}(\check{\mathbf{y}}|\mathbf{x})} \min_{\hat{P}(\hat{\mathbf{y}}|\mathbf{x})} \mathbb{E}_{\mathbf{X},\mathbf{Y}\sim\tilde{P};\hat{\mathbf{Y}}|\mathbf{X}\sim\hat{P};\check{\mathbf{Y}}|\mathbf{X}\sim\check{P}} \Big[\text{loss}(\hat{\mathbf{Y}},\check{\mathbf{Y}}) + \theta^{\mathrm{T}}\left(\Phi(\mathbf{X},\check{\mathbf{Y}}) - \Phi(\mathbf{X},\mathbf{Y})\right)\Big] \tag{3.8}$$

$$\overset{(d)}{=} \min_{\theta} \mathbb{E}_{\mathbf{X},\mathbf{Y}\sim\tilde{P}} \max_{\check{P}(\check{\mathbf{y}}|\mathbf{x})} \min_{\hat{P}(\hat{\mathbf{y}}|\mathbf{x})} \mathbb{E}_{\hat{\mathbf{Y}}|\mathbf{X}\sim\hat{P};\check{\mathbf{Y}}|\mathbf{X}\sim\check{P}} \Big[\text{loss}(\hat{\mathbf{Y}},\check{\mathbf{Y}}) + \theta^{\mathrm{T}}\left(\Phi(\mathbf{X},\check{\mathbf{Y}}) - \Phi(\mathbf{X},\mathbf{Y})\right)\Big] \tag{3.9}$$

$$\overset{(e)}{=} \min_{\theta_e,\theta_v} \mathbb{E}_{\mathbf{X},\mathbf{Y}\sim\tilde{P}} \max_{\check{P}(\check{\mathbf{y}}|\mathbf{x})} \min_{\hat{P}(\hat{\mathbf{y}}|\mathbf{x})} \mathbb{E}_{\hat{\mathbf{Y}}|\mathbf{X}\sim\hat{P};\check{\mathbf{Y}}|\mathbf{X}\sim\check{P}} \Big[\textstyle\sum_i^n \text{loss}(\hat{Y}_i,\check{Y}_i) \tag{3.10}$$

$$+ \theta_e \cdot \textstyle\sum_{(i,j)\in E}\left[\phi(\mathbf{X},\check{Y}_i,\check{Y}_j) - \phi(\mathbf{X},Y_i,Y_j)\right] + \theta_v \cdot \sum_i^n \left[\phi(\mathbf{X},\check{Y}_i) - \phi(\mathbf{X},Y_i)\right] \Big]$$

$$\overset{(f)}{=} \min_{\theta_e,\theta_v} \mathbb{E}_{\mathbf{X},\mathbf{Y}\sim\tilde{P}} \max_{\check{P}(\check{\mathbf{y}}|\mathbf{x})} \min_{\hat{P}(\hat{\mathbf{y}}|\mathbf{x})} \sum_{\hat{\mathbf{y}},\check{\mathbf{y}}} \hat{P}(\hat{\mathbf{y}}|\mathbf{x})\check{P}(\check{\mathbf{y}}|\mathbf{x}) \Big[\textstyle\sum_i^n \text{loss}(\hat{y}_i,\check{y}_i) \tag{3.11}$$

$$+ \theta_e \cdot \textstyle\sum_{(i,j)\in E}\left[\phi(\mathbf{x},\check{y}_i,\check{y}_j) - \phi(\mathbf{x},y_i,y_j)\right] + \theta_v \cdot \sum_i^n \left[\phi(\mathbf{x},\check{y}_i) - \phi(\mathbf{x},y_i)\right] \Big]$$

$$\overset{(g)}{=} \min_{\theta_e,\theta_v} \mathbb{E}_{\mathbf{X},\mathbf{Y}\sim\tilde{P}} \max_{\check{P}(\check{\mathbf{y}}|\mathbf{x})} \min_{\hat{P}(\hat{\mathbf{y}}|\mathbf{x})} \Big[\textstyle\sum_i^n \sum_{\hat{y}_i,\check{y}_i} \hat{P}(\hat{y}_i|\mathbf{x})\check{P}(\check{y}_i|\mathbf{x})\text{loss}(\hat{y}_i,\check{y}_i) \tag{3.12}$$

$$+ \textstyle\sum_{(i,j)\in E}\sum_{\check{y}_i,\check{y}_j} \check{P}(\check{y}_i,\check{y}_j|\mathbf{x})\left[\theta_e \cdot \phi(\mathbf{x},\check{y}_i,\check{y}_j)\right] - \sum_{(i,j)\in E}\theta_e \cdot \phi(\mathbf{x},y_i,y_j)$$

$$+ \textstyle\sum_i^n \sum_{\check{y}_i} \check{P}(\check{y}_i|\mathbf{x})\ \left[\theta_v \cdot \phi(\mathbf{x},\check{y}_i)\right] - \sum_i^n \theta_v \cdot \phi(\mathbf{x},y_i)\Big].$$

The transformation steps above are described as follows:

(a-d) We follow the similar transformation steps in the proof of Theorem 2.1.

(e) We apply our description of loss metrics which is additively decomposable into the loss for each node, and the features that can be decomposed into node and edge features. We also separate the notation for the Lagrange dual variable into the variable for the constraints on node features (θ_v) and and the variable for the edge features (θ_e).

(f) We rewrite the expectation over $\hat{P}(\hat{\mathbf{y}}|\mathbf{x})$ and $\check{P}(\check{\mathbf{y}}|\mathbf{x})$ in terms of the probability-weighted average.

(g) Based on the property of the loss metrics and feature functions, the sum over the exponentially many possibilities of $\hat{\mathbf{y}}$ and $\check{\mathbf{y}}$ can be simplified into the sum over individual nodes and edges values, resulting in the optimization over the node and edge marginal distributions.

□

Note that the optimization in Equation (3.4) over the node and edge marginal distributions resembles the optimization of CRFs (Sutton et al., 2012). In terms of computational complexity, this means that for a general graphical structure, the optimization above may be intractable. We focus on families of graphical structures in which the optimization is known to be tractable. In the next subsection, we begin with the case of tree-structured graphical models and then proceed with the case of graphical models with low treewidth. In both cases, we formulate the corresponding efficient learning algorithms.

3.3.2 Optimization

We first introduce our vector and matrix notations for AGM optimization. Without loss of generality, we assume the number of class labels k to be the same for all predicted variables $y_i, \forall i \in \{1, \ldots, n\}$. Let $\mathbf{p}_i$ be a vector with length k, where its a-th element contains $\hat{P}(\hat{y}_i = a|\mathbf{x})$, and let $\mathbf{Q}_{i,j}$ be a k-by-k matrix with its (a, b)-th cells store $\check{P}(\check{y}_i = a, \check{y}_j = b|\mathbf{x})$. We also use a vector and matrix notation to represent the ground truth label by letting $\mathbf{z}_i$ be a one-hot vector where its a-th element $\mathbf{z}_i^{(a)} = 1$ if $y_i = a$ or otherwise 0, and letting $\mathbf{Z}_{i,j}$ be a one-hot matrix where its (a, b)-th cell $\mathbf{Z}_{i,j}^{(a,b)} = 1$ if $y_i = a \wedge y_j = b$ or otherwise 0. For each node feature $\phi_l(\mathbf{x}, y_i)$, we denote $\mathbf{w}_{i,l}$ as a length k vector where its a-th element contains the value of $\phi_l(\mathbf{x}, y_i = a)$. Similarly, for each edge feature $\phi_l(\mathbf{x}, y_i, y_j)$, we denote $\mathbf{W}_{i,j,l}$ as a k-by-k matrix where its (a, b)-th cell contains the value of $\phi_l(\mathbf{x}, y_i = a, y_j = b)$. For a pairwise graphical model with tree structure, we rewrite Equation (3.4) using our vector and matrix notation with local marginal consistency constraints as follows:

$$\min_{\theta_e, \theta_v} \mathbb{E}_{\mathbf{X}, \mathbf{Y} \sim \tilde{P}} \max_{\mathbf{Q} \in \Delta} \min_{\mathbf{p} \in \Delta} \sum_i^n \Big[\mathbf{p}_i \mathbf{L}_i (\mathbf{Q}_{pt(i);i}^{\mathrm{T}} \mathbf{1}) + \Big\langle \mathbf{Q}_{pt(i);i} - \mathbf{Z}_{pt(i);i}, \textstyle\sum_l \theta_e^{(l)} \mathbf{W}_{pt(i);i;l} \Big\rangle \tag{3.13}$$

$$+ (\mathbf{Q}_{pt(i);i}^{\mathrm{T}} \mathbf{1} - \mathbf{z}_i)^{\mathrm{T}} (\textstyle\sum_l \theta_v^{(l)} \mathbf{w}_{i;l}) \Big]$$

$$\text{subject to: } \mathbf{Q}_{pt(pt(i));pt(i)}^{\mathrm{T}} \mathbf{1} = \mathbf{Q}_{pt(i);i} \mathbf{1}, \ \forall i \in \{1, \ldots, n\},$$

where $pt(i)$ indicates the parent of node i in the tree structure, $\mathbf{L}_i$ stores a loss matrix corresponding to the portion of the loss metric for node i, i.e., $\mathbf{L}_i^{(a,b)} = \text{loss}(\hat{y}_i = a, \check{y}_i = b)$, and $\langle \cdot, \cdot \rangle$ denotes the Frobenius inner product between two matrices, i.e., $\langle A, B \rangle = \sum_{i,j} A_{i,j} B_{i,j}$.

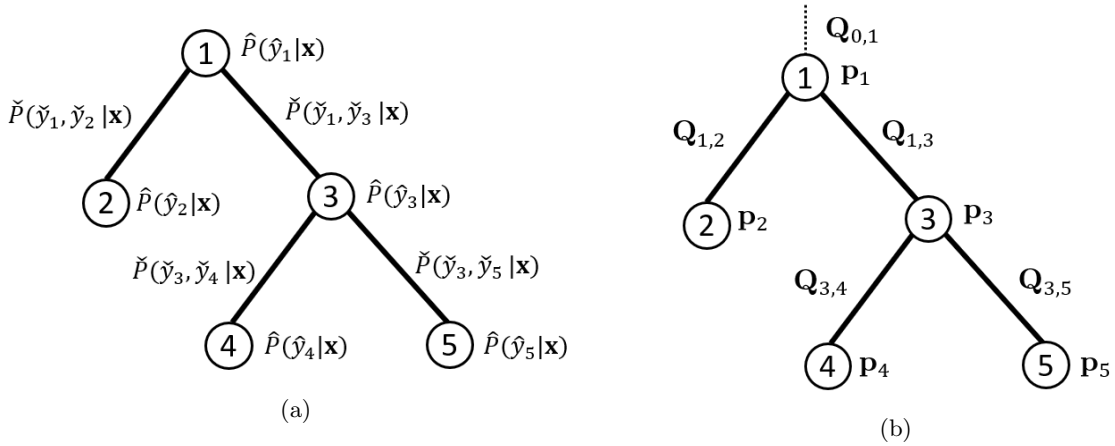


Figure 11. An example tree structure with five nodes and four edges with the corresponding marginal probabilities for predictor and adversary (a); and the matrix and vector notations of the probabilities (b). Note that we introduce a dummy edge variable on top of the root node to match the marginal constraints.

Note that we also employ probability simplex constraints (Δ) to each $\mathbf{Q}_{pt(i);i}$ and $\mathbf{p}_i$. Figure 11 shows an example tree structure with its marginal probabilities and the matrix notation of the probabilities.

3.3.2.1 Learning algorithm

We first focus on solving the inner minimax optimization of Equation (3.13). To simplify our notation, we denote the edge potentials $\mathbf{B}_{pt(i);i} = \sum_l \theta_e^{(l)} \mathbf{W}_{pt(i);i;l}$ and the node potentials $\mathbf{b}_i = \sum_l \theta_v^{(l)} \mathbf{w}_{i;l}$. We then rewrite the inner optimization of Equation (3.13) as:

$$\max_{\mathbf{Q}\in\Delta} \min_{\mathbf{p}\in\Delta} \sum_i^n \left[\mathbf{p}_i \mathbf{L}_i (\mathbf{Q}_{pt(i);i}^{\mathrm{T}} \mathbf{1}) + \left\langle \mathbf{Q}_{pt(i);i}, \mathbf{B}_{pt(i);i} \right\rangle + (\mathbf{Q}_{pt(i);i}^{\mathrm{T}} \mathbf{1})^{\mathrm{T}} \mathbf{b}_i \right] \tag{3.14}$$

$$\text{subject to: } \mathbf{Q}_{pt(pt(i));pt(i)}^{\mathrm{T}} \mathbf{1} = \mathbf{Q}_{pt(i);i} \mathbf{1}, \ \forall i \in \{1, \ldots, n\}.$$

To solve the optimization above, we use dual decomposition technique (Boyd et al., 2008; Sontag et al., 2011) that decompose the dual version of the optimization problem into several sub-problem that can be solved independently. By introducing the Lagrange variable $\mathbf{u}$ for the local marginal consistency constraint, we formulate an equivalent dual unconstrained optimization problem as shown in Theorem 3.2.

Theorem 3.2. *The constrained optimization in Equation* (3.14) *is equivalent to an unconstrained Lagrange dual problem with an inner optimization that can be solved independently for each node as follows:*

$$\min_{\mathbf{u}} \sum_{i}^{n} \left[\max_{\mathbf{Q}_i \in \Delta} \left\langle \mathbf{Q}_{pt(i);i}, \mathbf{B}_{pt(i);i} + \mathbf{1}\mathbf{b}_i^{\mathrm{T}} - \mathbf{u}_i \mathbf{1}^{\mathrm{T}} + \textstyle\sum_{k \in ch(i)} \mathbf{1}\mathbf{u}_k^{\mathrm{T}} \right\rangle + \min_{\mathbf{p}_i \in \Delta} \mathbf{p}_i \mathbf{L}_i (\mathbf{Q}_{pt(i);i}^{\mathrm{T}} \mathbf{1}) \right], \quad (3.15)$$

where $\mathbf{u}_i$ *is the Lagrange dual variable associated with the marginal constraint of* $\mathbf{Q}_{pt(pt(i));pt(i)}^{\mathrm{T}} \mathbf{1} = \mathbf{Q}_{pt(i);i} \mathbf{1}$*, and* $ch(i)$ *represent the children of node* i*.*

Proof.

$$\max_{\mathbf{Q}\in\Delta}\min_{\mathbf{p}\in\Delta}\sum_{i}^{n}\left[\mathbf{p}_i\mathbf{L}_i(\mathbf{Q}_{pt(i);i}^{\mathrm{T}}\mathbf{1})+\left\langle\mathbf{Q}_{pt(i);i},\mathbf{B}_{pt(i);i}\right\rangle+(\mathbf{Q}_{pt(i);i}^{\mathrm{T}}\mathbf{1})^{\mathrm{T}}\mathbf{b}_i\right] \tag{3.16}$$

$$\text{subject to: } \mathbf{Q}_{pt(pt(i));pt(i)}^{\mathrm{T}}\mathbf{1}=\mathbf{Q}_{pt(i);i}\mathbf{1},\ \forall i\in\{1,\ldots,n\}$$

$$\overset{(a)}{=}\max_{\mathbf{Q}\in\Delta}\min_{\mathbf{u}}\min_{\mathbf{p}\in\Delta}\sum_{i}^{n}\left[\mathbf{p}_i\mathbf{L}_i(\mathbf{Q}_{pt(i);i}^{\mathrm{T}}\mathbf{1})+\left\langle\mathbf{Q}_{pt(i);i},\mathbf{B}_{pt(i);i}\right\rangle+(\mathbf{Q}_{pt(i);i}^{\mathrm{T}}\mathbf{1})^{\mathrm{T}}\mathbf{b}_i\right] \tag{3.17}$$
$$+\sum_{i}^{n}\mathbf{u}_i^{\mathrm{T}}\left(\mathbf{Q}_{pt(pt(i));pt(i)}^{\mathrm{T}}\mathbf{1}-\mathbf{Q}_{pt(i);i}\mathbf{1}\right)$$

$$\overset{(b)}{=}\min_{\mathbf{u}}\max_{\mathbf{Q}\in\Delta}\min_{\mathbf{p}\in\Delta}\sum_{i}^{n}\left[\mathbf{p}_i\mathbf{L}_i(\mathbf{Q}_{pt(i);i}^{\mathrm{T}}\mathbf{1})+\left\langle\mathbf{Q}_{pt(i);i},\mathbf{B}_{pt(i);i}\right\rangle+(\mathbf{Q}_{pt(i);i}^{\mathrm{T}}\mathbf{1})^{\mathrm{T}}\mathbf{b}_i\right] \tag{3.18}$$
$$+\sum_{i}^{n}\mathbf{u}_i^{\mathrm{T}}\left(\mathbf{Q}_{pt(pt(i));pt(i)}^{\mathrm{T}}\mathbf{1}-\mathbf{Q}_{pt(i);i}\mathbf{1}\right)$$

$$\overset{(c)}{=}\min_{\mathbf{u}}\max_{\mathbf{Q}\in\Delta}\min_{\mathbf{p}\in\Delta}\sum_{i}^{n}\left[\mathbf{p}_i\mathbf{L}_i(\mathbf{Q}_{pt(i);i}^{\mathrm{T}}\mathbf{1})+\left\langle\mathbf{Q}_{pt(i);i},\mathbf{B}_{pt(i);i}\right\rangle+\left\langle\mathbf{Q}_{pt(i);i},\mathbf{1}\mathbf{b}_i^{\mathrm{T}}\right\rangle\right] \tag{3.19}$$
$$+\sum_{i}^{n}\left[\left\langle\mathbf{Q}_{pt(pt(i));pt(i)},\mathbf{1}\mathbf{u}_i^{\mathrm{T}}\right\rangle-\left\langle\mathbf{Q}_{pt(i);i},\mathbf{u}_i\mathbf{1}^{\mathrm{T}}\right\rangle\right]$$

$$\overset{(d)}{=}\min_{\mathbf{u}}\max_{\mathbf{Q}\in\Delta}\min_{\mathbf{p}\in\Delta}\sum_{i}^{n}\left[\mathbf{p}_i\mathbf{L}_i(\mathbf{Q}_{pt(i);i}^{\mathrm{T}}\mathbf{1})+\left\langle\mathbf{Q}_{pt(i);i},\mathbf{B}_{pt(i);i}+\mathbf{1}\mathbf{b}_i^{\mathrm{T}}-\mathbf{u}_i\mathbf{1}^{\mathrm{T}}+\textstyle\sum_{k\in ch(i)}\mathbf{1}\mathbf{u}_k^{\mathrm{T}}\right\rangle\right]. \tag{3.20}$$

The transformation steps above are described as follows:

(a) We introduce the Lagrange dual variable $\mathbf{u}$, where $\mathbf{u}_i$ is the dual variable associated with the marginal constraint of $\mathbf{Q}_{pt(pt(i));pt(i)}^{\mathrm{T}}\mathbf{1}=\mathbf{Q}_{pt(i);i}\mathbf{1}$.

(b) Similar to the analysis in Theorem 1, strong duality holds due to Sion's minimax theorem. Therefore, we can flip the optimization order of $\mathbf{Q}$ and $\mathbf{u}$.

(c) We rewrite the vector multiplication over $\mathbf{Q}_{pt(i);i}\mathbf{1}$ or $\mathbf{Q}_{pt(i);i}^{\mathrm{T}}\mathbf{1}$ with the corresponding Frobenius inner product notations.

(d) We regroup the terms in the optimization above by considering the parent-child relations in the tree for each node. Note that $ch(i)$ represents the children of node i.

□

We denote matrix $\mathbf{A}_{pt(i);i} \triangleq \mathbf{B}_{pt(i);i} + \mathbf{1}\mathbf{b}_i^{\mathrm{T}} - \mathbf{u}_i\mathbf{1}^{\mathrm{T}} + \sum_{k \in ch(i)} \mathbf{1}\mathbf{u}_k^{\mathrm{T}}$ to simplify the inner optimization in Equation (3.15). Let us define $\mathbf{r}_i \triangleq \mathbf{Q}_{pt(i);i}^{\mathrm{T}}\mathbf{1}$ and $\mathbf{a}_i$ be the column wise maximum of matrix $\mathbf{A}_{pt(i);i}$, i.e., $\mathbf{a}_i^{(l)} = \max_l \mathbf{A}_{l;i}$. Given the value of $\mathbf{u}$, each of the inner optimizations in Equation (3.15) can be equivalently solved in terms of our newly defined variable changes $\mathbf{r}_i$ and $\mathbf{a}_i$ as follows:

$$\max_{\mathbf{r}_i \in \Delta} \left[\mathbf{a}_i^{\mathrm{T}}\mathbf{r}_i + \min_{\mathbf{p}_i \in \Delta} \mathbf{p}_i\mathbf{L}_i\mathbf{r}_i \right]. \tag{3.21}$$

Note that this resembles the optimization in a standard adversarial multiclass classification problem we discussed in Section 2.4, i.e., Equation (2.13) with $\mathbf{L}_i$ as the loss matrix and $\mathbf{a}_i$ as the class-based potential vector, without the potential for the true label. As discussed in Section 2.4, Equation (3.21) can be solved analytically for several forms of loss metrics (e.g., zero-one, absolute, squared, abstention loss metrics), or as a linear program for a more general loss metrics. Given the solution of this inner optimization, we use a sub-gradient based optimization to find the optimal Lagrange dual variables $\mathbf{u}^*$.

To recover our original variables for the adversary's marginal distribution $\mathbf{Q}^*_{pt(i);i}$ given the optimal dual variables $\mathbf{u}^*$, we use the following steps. First, we use $\mathbf{u}^*$ and Equation (3.21) to compute the value of the node marginal probability $\mathbf{r}_i^*$. With the additional information that

we know the value of $\mathbf{r}_i^*$ (i.e., the adversary's node probability), Equation (3.14) can be solved independently for each $\mathbf{Q}_{pt(i);i}$ to obtain the optimal $\mathbf{Q}^*_{pt(i);i}$ as follows:

$$\mathbf{Q}^*_{pt(i);i} = \underset{\mathbf{Q}_{pt(i);i} \in \Delta}{\operatorname{argmax}} \left\langle \mathbf{Q}_{pt(i);i}, \mathbf{B}_{pt(i);i} \right\rangle \text{ subject to: } \mathbf{Q}^{\mathrm{T}}_{pt(i);i}\mathbf{1} = \mathbf{r}^*_i,\ \mathbf{Q}_{pt(i);i}\mathbf{1} = \mathbf{r}^*_{pt(i)}. \tag{3.22}$$

Note that the optimization above resembles an optimal transport problem over two discrete distributions (Villani, 2008) with cost matrix $-\mathbf{B}_{pt(i);i}$. This optimal transport problem can be solved using a linear program solver or a more sophisticated solver (e.g., using Sinkhorn distances (Cuturi, 2013)).

For our overall learning algorithm, we use the optimal adversary's marginal distributions $\mathbf{Q}^*_{pt(i);i}$ to compute the sub-differential of the AGM formulation (Equation (3.13)) with respect to θ_v and θ_e. The sub-differential for $\theta_v^{(l)}$ includes the expected node feature difference $\mathbb{E}_{\mathbf{X},\mathbf{Y}\sim\tilde{P}} \sum_i^n (\mathbf{Q}^{*\mathrm{T}}_{pt(i);i}\mathbf{1} - \mathbf{z}_i)^{\mathrm{T}} \mathbf{w}_{i;l}$, whereas the sub-differential for $\theta_e^{(l)}$ includes the expected edge feature difference $\mathbb{E}_{\mathbf{X},\mathbf{Y}\sim\tilde{P}} \sum_i^n \left\langle \mathbf{Q}^*_{pt(i);i} - \mathbf{Z}_{pt(i);i}, \mathbf{W}_{pt(i);i;l} \right\rangle$. Using this sub-differential information, we employ a stochastic sub-gradient based algorithm to obtain the optimal θ_v^* and θ_e^*.

3.3.2.2 Prediction algorithms

We propose two different prediction schemes: probabilistic and non-probabilistic prediction.

Probabilistic prediction. Our probabilistic prediction is based on the predictor's label probability distribution in the adversarial prediction formulation. Given fixed values of θ_v and

θ_e, we solve a minimax optimization similar to Equation (3.13) by flipping the order of the predictor and adversary distribution as follows:

$$\min_{\mathbf{p}\in\boldsymbol{\Delta}} \max_{\mathbf{Q}\in\boldsymbol{\Delta}} \sum_{i}^{n} \left[\mathbf{p}_i \mathbf{L}_i (\mathbf{Q}_{pt(i);i}^{\mathrm{T}} \mathbf{1}) + \left\langle \mathbf{Q}_{pt(i);i}, \textstyle\sum_l \theta_e^{(l)} \mathbf{W}_{pt(i);i;l} \right\rangle + (\mathbf{Q}_{pt(i);i}^{\mathrm{T}} \mathbf{1})^{\mathrm{T}} (\textstyle\sum_l \theta_v^{(l)} \mathbf{w}_{i;l}) \right] \quad (3.23)$$

$$\text{subject to: } \mathbf{Q}_{pt(pt(i));pt(i)}^{\mathrm{T}} \mathbf{1} = \mathbf{Q}_{pt(i);i} \mathbf{1}, \ \forall i \in \{1, \ldots, n\}.$$

To solve the inner maximization of $\mathbf{Q}$ we use a similar technique as in MAP inference for CRFs. We then use a projected gradient optimization technique to solve the outer minimization over $\mathbf{p}$ and a technique for projecting to the probability simplex (Duchi et al., 2008).

Non-probabilistic prediction. Our non-probabilistic prediction scheme is similar to SSVM's prediction algorithm. In this scheme, we find $\hat{\mathbf{y}}$ that maximizes the potential value, i.e., $\hat{\mathbf{y}} = \text{argmax}_{\mathbf{y}} f(\mathbf{x}, \mathbf{y})$, where $f(\mathbf{x}, \mathbf{y}) = \theta^{\mathrm{T}} \Phi(\mathbf{x}, \mathbf{y})$. This prediction scheme is faster than the probabilistic scheme since we only need a single run of a Viterbi-like algorithm for tree structures.

3.3.2.3 Runtime analysis

Each stochastic update in our algorithm involves finding the optimal $\mathbf{u}$ and recovering the optimal $\mathbf{Q}$ to be used in a sub-gradient update. Each iteration of a sub-gradient based optimization to solve $\mathbf{u}$ costs $\mathcal{O}(n \cdot c(\mathbf{L}))$ time where n is the number of nodes and $c(\mathbf{L})$ is the cost for solving the optimization in Equation (3.21) for the loss matrix $\mathbf{L}$. Recovering all of the adversary's marginal distributions $\mathbf{Q}_{pt(i);i}$ using a fast Sinkhorn distance solver has the empirical complexity of $\mathcal{O}(nk^2)$ where k is the number of classes (Cuturi, 2013). The total running time

of our method depends on the loss metric we use. For example, if the loss metric is the additive zero-one loss, the total complexity of one stochastic gradient update is $\mathcal{O}(nlk \log k + nk^2)$ time, where l is the number of iterations needed to obtain the optimal $\mathbf{u}$ and $\mathcal{O}(k \log k)$ time is the cost for solving Equation (3.21) for the zero-one loss (Fathony et al., 2016). In practice, we find the average value of l to be relatively small. This runtime complexity is competitive with the CRF, which requires $\mathcal{O}(nk^2)$ time to perform message-passing over a tree to compute the marginal distribution of each parameter update, and also with structured SVM where each iteration requires computing the most violated constraint, which also costs $\mathcal{O}(nk^2)$ time for running a Viterbi-like algorithm over a tree structure.

3.3.2.4 Learning algorithm for graphical structure with low treewidth

Our algorithm for tree-based graphs can be easily extended to the case of graphical structures with low treewidth. Similar to the case of the junction tree algorithm for probabilistic graphical models, we first construct a junction tree representation for the graphical structure. We then solve a similar optimization as in Equation (3.13) on the junction tree. In this case, the time complexity of one stochastic gradient update of the algorithm is $\mathcal{O}(nlwk^{(w+1)} \log k + nk^{2(w+1)})$ time for the optimization with an additive zero-one loss metric, where n is the number of cliques in the junction tree, k is the number of classes, l is the number of iterations in the inner optimization, and w is the treewidth of the graph. This time complexity is competitive with the time complexities of CRF and SSVM which are also exponential in the treewidth of the graph.

3.3.3 Fisher consistency analysis

A key theoretical advantage of our approach over the structured SVM is that it provides Fisher consistency. This guarantees that under the true distribution $P(\mathbf{x}, \mathbf{y})$, the learning algorithm yields a Bayes optimal prediction with respect to the loss metric (Tewari and Bartlett, 2007; Liu, 2007). In this setting, the learning algorithm is allowed to optimize over all measurable functions, or similarly, it has a feature representation of unlimited richness. We establish the Fisher consistency of our AGM approach in Theorem 3.3.

Theorem 3.3. *The AGM approach is Fisher consistent for all additive loss metrics.*

Proof. As established in Theorem 3.1, pairwise marginal probabilities are sufficient statistics of the adversary's distribution. An unlimited access to arbitrary rich feature representation constrains the adversary's distribution in Equation (3.3) to match the marginal probabilities of the true distribution, making the optimization in Equation (3.3) equivalent to $\min_{\hat{\mathbf{y}}} \mathbb{E}_{\mathbf{X},\mathbf{Y}\sim P}\left[\text{loss}(\hat{\mathbf{y}}, \mathbf{Y})\right]$, which is the Bayes optimal prediction for the loss metric. □

3.4 Experimental Evaluations

To evaluate our approach, we apply AGM to two different tasks: predicting emotion intensity from a sequence of images, and labeling entities in parse trees with semantic roles. We show the benefit of our method compared with a conditional random field (CRF) and a structured SVM (SSVM).

3.4.1 Facial emotion intensity prediction

We evaluate our approach in the facial emotion intensity prediction task (Kim and Pavlovic, 2010). Given a sequence of facial images, the task is to predict the emotion intensity for each individual image. The emotion intensity labels are categorized into three ordinal categories: `neutral` < `increasing` < `apex`, reflecting the degree of intensity. The dataset contains 167 sequences collected from 100 subjects consisting of six types of basic emotions (anger, disgust, fear, happiness, sadness, and surprise). In terms of the features used for prediction, we follow an existing feature extraction procedure (Kim and Pavlovic, 2010) that uses Haar-like features and the PCA algorithm to reduce the feature dimensionality.

In our experimental setup, we combine the data from all six different emotions and focus on predicting the ordinal category of emotion intensity. From the whole 167 sequences, we construct 20 different random splits of the training and the testing datasets with 120 sequences of training samples and 47 sequences of testing samples. We use the training set in the first split to perform cross validation to obtain the best regularization parameters and then use the best parameter in the evaluation phase for all 20 different splits of the dataset.

In the evaluation, we use six different loss metrics. The first three metrics are the average of zero-one, absolute and squared loss metrics for each node in the graph (where we assign label values: `neutral` = 1, `increasing` = 2, and `apex` = 3). The other three metrics are the weighted version of the zero-one, absolute and squared loss metrics. These weighted variants of the loss metrics reflect the focus on the prediction task by emphasizing the prediction on

particular nodes in the graph. In this experiment, we set the weight to be the position in the sequence so that we focus more on the latest nodes in the sequences.

We compare our method with CRF and SSVM models. Both the AGM and the SSVM can incorporate the task's customized loss metrics in the learning process. The prediction for AGM and SSVM is done by taking an arg-max of potential values, i.e., $\text{argmax}_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}) = \theta \cdot \Phi(\mathbf{x}, \mathbf{y})$. For CRF, the training step aims to model the conditional probability $\hat{P}_\theta(\mathbf{y}|\mathbf{x})$. The CRF's predictions are computed using the Bayes optimal prediction with respect to the loss metric and CRF's conditional probability, i.e., $\text{argmin}_{\mathbf{y}} \mathbb{E}_{\hat{\mathbf{Y}}|\mathbf{x}\sim\hat{P}_\theta}[\text{loss}(\mathbf{y}, \hat{\mathbf{Y}})]$.

We report the loss metrics averaged over the dataset splits as shown in Table VII. We highlight the result that is either the best result or not significantly worse than the best result (using Wilcoxon signed-rank test with $\alpha = 0.05$). The result shows that our method significantly outperforms CRF in three cases (absolute, weighted zero-one, and weighted absolute losses), and statistically ties with CRF in one case (squared loss), while only being outperformed by CRF in one case (zero-one loss). AGM also outperforms SSVM in three cases (absolute, squared, and weighted zero-one losses), and statistically ties with SSVM in one case (weighted absolute loss), while only being outperformed by SSVM in one case (weighted squared loss). In the overall result, AGM maintains advantages compared to CRFs and SSVMs in both the overall average loss and the number of "indistinguishably best" performances on all cases. These results may reflect the theoretical benefit that AGM has over CRF and SSVM mentioned in Section 3 when learning from noisy labels.

TABLE VII. The average loss metrics for the emotion intensity prediction. Bold numbers indicate the best or not significantly worse than the best results (Wilcoxon signed-rank test with $\alpha = 0.05$).

Loss metrics	AGM	CRF	SSVM
zero-one, unweighted	0.34	**0.32**	0.37
absolute, unweighted	**0.33**	0.34	0.40
squared, unweighted	**0.38**	**0.38**	0.40
zero-one, weighted	**0.28**	0.32	0.29
absolute, weighted	**0.29**	0.36	**0.29**
squared, weighted	0.36	0.40	**0.33**
average	0.33	0.35	0.35
# bold	4	2	2

3.4.2 Semantic role labeling

We evaluate the performance of our algorithm on the semantic role labeling task for the CoNLL 2005 dataset (Carreras and Màrquez, 2005). Given a sentence and its syntactic parse tree as the input, the task is to recognize the semantic role of each constituent in the sentence as propositions expressed by some target verbs in the sentence. There are a total of 36 semantic roles grouped by their types of: numbered arguments, adjuncts, references of numbered and adjunct arguments, continuation of each class type and the verb. We prune the syntactic trees according to (Xue and Palmer, 2004), i.e., we only include siblings of the nodes which are on the path from the verb (V) to the root and also the immediate children in case that the node is a propositional phrase (PP). Following the setup used by (Cohn and Blunsom, 2005), we extract the same syntactic and contextual features and label non-argument constituents and children

nodes of arguments as "outside" (O). Additionally, in our experiment we simplify the prediction task by reducing the number of labels. Specifically, we choose the three most common labels in the WSJ test dataset, i.e., A0,A1,A2 and their references R-A0,R-A1,R-A2, and we combine the rest of the classes as one separate class R. Thus, together with outside O and verb V, we have a total of nine classes in our experiment.

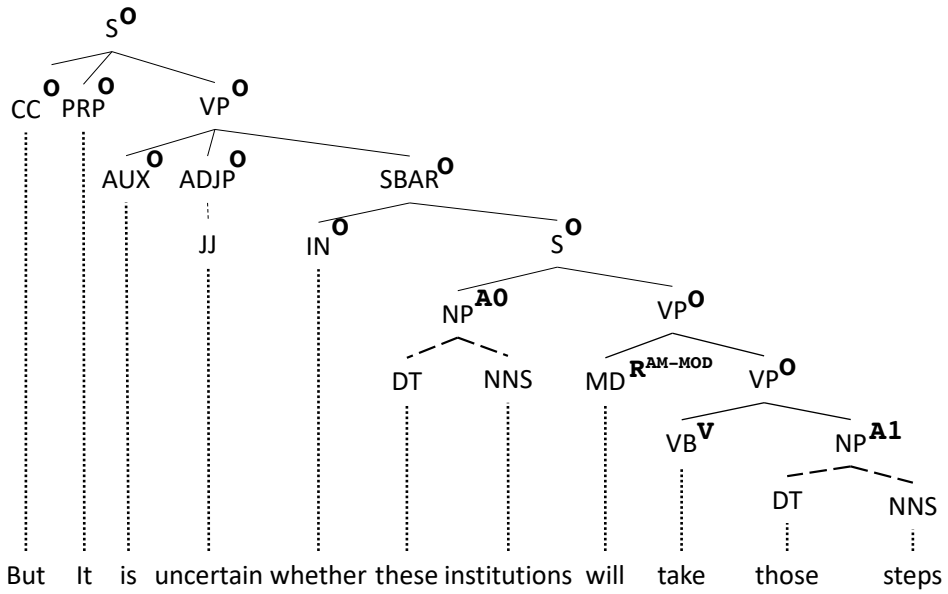


Figure 12. Example of a syntax tree with semantic role labels as bold superscripts. The dotted and dashed lines show the pruned edges from the tree. The original label AM-MOD is among class R in our experimental setup.

In the evaluation, we use a cost-sensitive loss matrix that reflects the importance of each label. We use the same cost-sensitive loss matrix to evaluate the prediction of all nodes in the

graph. The cost-sensitive loss matrix is constructed by picking a random order of the class label and assigning an ordinal loss based on the order of the labels. We compare the average cost-sensitive loss metric of our method with the CRF and the SSVM as shown in Table VIII. As we can see from the table, our result is competitive with SSVM, while maintaining an advantage over the CRF. This experiment shows that incorporating customized losses into the training process of learning algorithms is important for some structured prediction tasks. Both the AGM and the SSVM are designed to align their learning algorithms with the customized loss metric, whereas CRF can only utilize the loss metric information in its prediction step.

TABLE VIII. The average loss metrics for the semantic role labeling task.

Loss metrics	AGM	CRF	SSVM
cost-sensitive loss	0.14	0.19	0.14

3.5 Conclusions and Future Works

We introduced adversarial graphical models, a robust approach to structured prediction that possesses the main benefits of existing methods: (1) it guarantees the same Fisher consistency possessed by conditional random fields (Lafferty et al., 2001); (2) it aligns the target loss metric with the learning objective, as in maximum margin methods (Joachims, 2005; Taskar et al., 2005a); and (3) its computational run time complexity is primarily shaped by the graph

treewidth, which is similar to both graphical modeling approaches. Our experimental results demonstrate the benefits of this approach on structured prediction tasks with low treewidth.

For more complex graphical structures with high treewidth, our proposed algorithm may not be efficient. Similar to the case of CRFs and SSVMs, approximation algorithms may be needed to solve the optimization in AGM formulations for these structures. In future work, we plan to investigate the optimization techniques and applicable approximation algorithms for general graphical structures.

CHAPTER 4

ADVERSARIAL BIPARTITE MATCHING IN GRAPHS

(This chapter was previously published as "Efficient and Consistent Adversarial Bipartite Matching" (Fathony et al., 2018a) in the Proceedings of the 35th International Conference on Machine Learning (ICML 2018).)

4.1 Introduction

How can the elements from two sets be paired one-to-one to have the largest sum of pairwise utilities? This *maximum weighted perfect bipartite matching* problem is a classical combinatorial optimization problem in computer science. It can be formulated and efficiently solved in polynomial time as a linear program or using more specialized *Hungarian algorithm* techniques (Kuhn, 1955). This has made it an attractive formalism for posing a wide range of problems, including recognizing correspondences in similar images (Belongie et al., 2002; Liu et al., 2008; Zhu et al., 2008; Rui et al., 2007), finding word alignments in text (Chan and Ng, 2008), and providing ranked lists of items for information retrieval tasks (Amini et al., 2008).

Machine learning methods seek to estimate the pairwise utilities of bipartite graphs so that the maximum weighted complete matching is most compatible with the (distribution of) ground truth matchings of training data. When these utilities are learned abstractly, they can be employed to make predictive matchings for test samples. Unfortunately, important measures of incompatibility (e.g., the Hamming loss) are often non-continuous with many local optima in

the predictors' parameter spaces, making direct minimization intractable. Given this difficulty, two natural desiderata for any predictor are:

- **Efficiency:** learning from training data and making predictions must be computed efficiently in (low-degree) polynomial time; and
- **Consistency:** the predictor's training objectives must also minimize the underlying Hamming loss, at least under ideal learning conditions (given the true distribution and fully expressive model parameters).

Existing methods for learning bipartite matchings fail in one or the other of these desiderata; exponentiated potential fields models (Lafferty et al., 2001; Petterson et al., 2009) are intractable for large sets of items, while maximum margin methods based on the hinge loss surrogate (Taskar et al., 2005a; Tsochantaridis et al., 2005) lack Fisher consistency (Tewari and Bartlett, 2007; Liu, 2007). We discuss these limitations formally in Section 4.2.

Given the deficiencies of the existing methods, we contribute the first approach for learning bipartite matchings that is both computationally efficient and Fisher consistent. Our approach is based on an adversarial formulation for learning (Topsøe, 1979; Grünwald and Dawid, 2004; Asif et al., 2015) that poses prediction-making as a data-constrained zero-sum game between a player seeking to minimize the expected loss and an adversarial data approximator seeking to maximize the expected loss. We present an efficient approach for solving the corresponding zero-sum game arising from our formulation by decomposing the game's solution into marginal probabilities and optimizes these marginal probabilities directly to obtain an equilibrium saddle

point for the game. We then establish the computational efficiency and consistency of this approach and demonstrate its benefits experimentally.

4.2 Previous Inefficiency and Inconsistency

4.2.1 Bipartite Matching Task

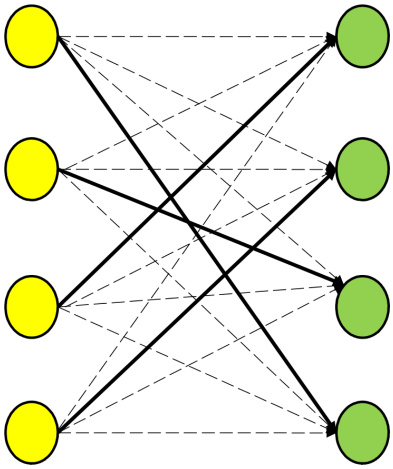


Figure 13. Bipartite matching task with n=4.

Given two sets of elements A and B of equal size ($|A| = |B|$), a maximum weighted bipartite matching π is the one-to-one mapping (e.g., Figure 13) from each element in A to each element in B that maximizes the sum of potentials: $\max_{\pi \in \Pi} \psi(\pi) = \max_{\pi \in \Pi} \sum_i \psi_i(\pi_i)$. Here $\pi_i \in [n] := \{1, 2, \ldots, n\}$ is the entry in B that is matched with the i-th entry of A. The set of possible solutions Π is simply all permutation of $[n]$. Many machine learning tasks pose

prediction as the solution to this problem, including: word alignment for natural language processing tasks (Taskar et al., 2005b; Padó and Lapata, 2006; MacCartney et al., 2008); learning correspondences between images in computer vision applications (Belongie et al., 2002; Dellaert et al., 2003); protein structure analysis in computational biology (Taylor, 2002; Wang et al., 2004); and learning to rank a set of items for information retrieval tasks (Dwork et al., 2001; Le and Smola, 2007). Thus, learning appropriate weights $\psi_i(\cdot)$ for bipartite graph matchings is a key problem for many application areas.

4.2.2 Performance Evaluation and Fisher Consistency

Given a predicted permutation, π', and the "ground truth" permutation, π, the **Hamming loss** counts the number of mistaken pairings: $\text{loss}_{\text{Ham}}(\pi, \pi') = \sum_{i=1}^{n} 1(\pi'_i \neq \pi_i)$, where $1(\cdot) = 1$ if $\cdot$ is true and 0 otherwise. When the "ground truth" is a distribution over permutations, $P(\pi)$, rather than a single permutation, the (set of) **Bayes optimal** prediction(s) is: $\text{argmin}_{\pi'} \sum_{\pi} P(\pi)\ \text{loss}_{\text{Ham}}(\pi, \pi')$. For a predictor to be **Fisher consistent**, it must provide a Bayes optimal prediction for any possible distribution $P(\pi)$ when trained from that exact distribution using the predictor's most general possible parameterization (e.g., all measurable functions ψ for potential-based models).

4.2.3 Exponential Family Random Field Approach

A probabilistic approach to learning bipartite graphs uses an exponential family distribution over permutations, $P_\psi(\pi) = e^{\sum_{i=1}^{n} \psi_i(\pi_i)} / Z_\psi$, trained by maximizing training data likelihood. This provides certain statistical consistency guarantees for its marginal probability estimates (Petterson et al., 2009). Specifically, if the potentials ψ are chosen from the

space of all measurable functions to maximize the likelihood of the true distribution of permutations $P(\pi)$, then $P_\psi(\pi)$ will match the marginal probabilities of the true distribution: $\forall i, j, P_\psi(\pi_i = j) = P(\pi_i = j)$. This implies Fisher consistency because the MAP estimate under this distribution, which can be obtained as a maximum weighted bipartite matching, is Bayes optimal.

The key challenge with this approach is its computational complexity. The normalization term, Z_ψ, is the permanent of a matrix defined in terms of exponentiated potential terms: $Z_\psi = \sum_\pi \prod_{i=1}^n e^{\psi_i(\pi_i)} = \text{perm}(\mathbf{M})$ where $M_{i,j} = e^{\psi_i(j)}$. For sets of small size (e.g., $n = 5$), enumerating the permutations is tractable and learning using the exponential random field model incurs a run-time cost that is acceptable in practice (Petterson et al., 2009). However, the matrix permanent computation is a *#P-hard* problem to compute exactly (Valiant, 1979). Monte Carlo sampling approaches are used instead of permutation enumeration to maximize the data likelihood (Petterson et al., 2009; Volkovs and Zemel, 2012). Though exact samples can be generated efficiently in polynomial time (Huber and Law, 2008), the number of samples needed for reliable likelihood or gradient estimates makes this approach infeasible for applications with even modestly-sized sets of $n = 20$ elements (Petterson et al., 2009).

In some applications such as word alignment, exponential family random field models that relax the permutation constraints in the label to the set of standard multiclass classifications (i.e., a class label can appear more than once in the prediction) were proposed to avoid the intractability of the normalization term computation (Blunsom and Cohn, 2006). In this case,

the model reduces to the standard linear chain conditional random fields (CRF) model (refer to Chapter 3 for a more discussion about the CRF).

4.2.4 Maximum Margin Approach

Maximum margin methods for structured prediction seek potentials ψ that minimize the training sample hinge loss:

$$\min_{\psi} \mathbb{E}_{\pi \sim \tilde{P}} \left[\max_{\pi'} \left\{ \text{loss}(\pi, \pi') + \psi(\pi') \right\} - \psi(\pi) \right], \tag{4.1}$$

where $\tilde{P}$ is the empirical distribution. Finding the optimal ψ is a convex optimization problem (Boyd and Vandenberghe, 2004) that can generally be tractably solved using constraint generation methods as long as the maximizing assignments can be found efficiently. In the case of permutation learning, finding the permutation π' with highest hinge loss reduces to a maximum weighted bipartite matching problem and can therefore be solved efficiently.

Though computationally efficient, maximum margin approaches for learning to make perfect bipartite matches lack **Fisher consistency**, which requires the prediction $\pi^* = \text{argmax}_{\pi} \psi(\pi)$ resulting from Equation (4.1) to minimize the expected risk, $\mathbb{E}_{\pi \sim \tilde{P}} [\text{loss}(\pi, \pi')]$, for all distributions $\tilde{P}$. We consider a distribution over permutations that is an extension of a counterexample for multiclass classification consistency analysis with no majority label (Liu, 2007): $P(\pi = [1\ 2\ 3]) = 0.4; P(\pi = [2\ 3\ 1]) = 0.3;$ and $P(\pi = [3\ 1\ 2]) = 0.3$. The potential function $\psi_i(j) = 1$ if $i = j$ and 0 otherwise, provides a Bayes optimal permutation prediction for this distribution and an expected hinge loss of $3.6 = 0.4(3 - 3) + 0.3(3 + 3) + 0.3(3 + 3)$. However,

the expected hinge loss is optimally minimized with a value of 3 when $\psi_i(j) = 0, \forall i, j$, which is indifferent between all permutations and is not Bayes optimal. Thus, Fisher consistency is not guaranteed.

4.3 Adversarial Bipartite Matching

To overcome the computational inefficiency of exponential random field methods and the Fisher inconsistency of maximum margin methods, we formulate the task of learning for bipartite matching problems as an adversarial structured prediction task. We then present an efficient approach for solving the resulting game over permutations.

4.3.1 Permutation Mixture Formulation

The training data for bipartite matching consists of triplets (A, B, π) where A and B are two sets of nodes with equal size and π is the assignment. To simplify the notation, we denote x as the bipartite graph containing the nodes A and B. We also denote $\phi(x, \pi)$ as a vector that enumerates the joint feature representations based on the bipartite graph x and the matching assignment π. This joint feature is defined additively over each node assignment, i.e., $\phi(x, \pi) = \sum_{i=1}^{n} \phi_i(x, \pi_i)$.

Our approach seeks a predictor that robustly minimizes the Hamming loss against the worst-case permutation mixture probability that is consistent with the statistics of the training data. In this setting, a predictor makes a probabilistic prediction over the set of all possible assignments (denoted as $\hat{P}$). Instead of evaluating the predictor with the empirical distribution, the predictor is pitted against an adversary that also makes a probabilistic prediction (denoted as $\check{P}$). The predictor's objective is to minimize the expected loss function calculated from the

predictor's and adversary's probabilistic predictions, while the adversary seeks to maximize the loss. The adversary (and only 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 joint features $\phi(x, \pi)$. Formally, we write our formulation as:

$$\min_{\hat{P}(\hat{\pi}|x)} \max_{\check{P}(\check{\pi}|x)} \mathbb{E}_{x\sim\tilde{P};\hat{\pi}|x\sim\hat{P};\check{\pi}|x\sim\check{P}} \left[\text{loss}(\hat{\pi}, \check{\pi})\right] \tag{4.2}$$
$$\text{subject to: } \mathbb{E}_{x\sim\tilde{P};\check{\pi}|x\sim\check{P}} \left[\sum_{i=1}^{n} \phi_i(x, \check{\pi}_i)\right] = \mathbb{E}_{(x,\pi)\sim\tilde{P}} \left[\sum_{i=1}^{n} \phi_i(x, \pi_i)\right].$$

Using the method of Lagrangian multipliers and strong duality for convex-concave saddle point problems (Von Neumann and Morgenstern, 1945; Sion, 1958), The optimization in Equation (4.2) can be equivalently solved in the dual formulation:

$$\min_{\theta} \mathbb{E}_{x,\pi\sim\tilde{P}} \min_{\hat{P}(\hat{\pi}|x)} \max_{\check{P}(\check{\pi}|x)} \mathbb{E}_{\substack{\hat{\pi}|x\sim\hat{P} \\ \check{\pi}|x\sim\check{P}}} \Bigg[\text{loss}(\hat{\pi}, \check{\pi}) + \theta \cdot \sum_{i=1}^{n} \left(\phi_i(x, \check{\pi}_i) - \phi_i(x, \pi_i)\right)\Bigg], \tag{4.3}$$

where θ is the Lagrange dual variable for the moment matching constraints. Below is the detailed step-by-step transformation from the primal mixture formulation of the adversarial prediction task for bipartite matching (Equation (4.2)) to the dual formulation (Equation (4.3)):

$$\min_{\hat{P}(\hat{\pi}|x)} \max_{\check{P}(\check{\pi}|x)} \mathbb{E}_{x\sim\tilde{P};\hat{\pi}|x\sim\hat{P};\check{\pi}|x\sim\check{P}} \left[\text{loss}(\hat{\pi},\check{\pi})\right] \tag{4.4}$$

$$\text{subject to:} \quad \mathbb{E}_{x\sim\tilde{P};\check{\pi}|x\sim\check{P}} \left[\sum_{i=1}^{n} \phi_i(x,\check{\pi}_i)\right] = \mathbb{E}_{(x,\pi)\sim\tilde{P}} \left[\sum_{i=1}^{n} \phi_i(x,\pi_i)\right]$$

$$= \max_{\check{P}(\check{\pi}|x)} \min_{\hat{P}(\hat{\pi}|x)} \mathbb{E}_{x\sim\tilde{P};\hat{\pi}|x\sim\hat{P};\check{\pi}|x\sim\check{P}} \left[\text{loss}(\hat{\pi},\check{\pi})\right] \tag{4.5}$$

$$\text{subject to:} \quad \mathbb{E}_{x\sim\tilde{P};\check{\pi}|x\sim\check{P}} \left[\sum_{i=1}^{n} \phi_i(x,\check{\pi}_i)\right] = \mathbb{E}_{(x,\pi)\sim\tilde{P}} \left[\sum_{i=1}^{n} \phi_i(x,\pi_i)\right]$$

$$= \max_{\check{P}(\check{\pi}|x)} \min_{\theta} \min_{\hat{P}(\hat{\pi}|x)} \mathbb{E}_{(x,\pi)\sim\tilde{P};\hat{\pi}|x\sim\hat{P};\check{\pi}|x\sim\check{P}} \left[\text{loss}(\hat{\pi},\check{\pi}) + \theta^{\mathrm{T}} \left(\sum_{i=1}^{n} \phi_i(x,\check{\pi}_i) - \sum_{i=1}^{n} \phi_i(x,\pi_i)\right)\right] \tag{4.6}$$

$$= \min_{\theta} \max_{\check{P}(\check{\pi}|x)} \min_{\hat{P}(\hat{\pi}|x)} \mathbb{E}_{(x,\pi)\sim\tilde{P};\hat{\pi}|x\sim\hat{P};\check{\pi}|x\sim\check{P}} \left[\text{loss}(\hat{\pi},\check{\pi}) + \theta^{\mathrm{T}} \left(\sum_{i=1}^{n} \phi_i(x,\check{\pi}_i) - \sum_{i=1}^{n} \phi_i(x,\pi_i)\right)\right] \tag{4.7}$$

$$= \min_{\theta} \mathbb{E}_{(x,\pi)\sim\tilde{P}} \max_{\check{P}(\check{\pi}|x)} \min_{\hat{P}(\hat{\pi}|x)} \mathbb{E}_{\hat{\pi}|x\sim\hat{P};\check{\pi}|x\sim\check{P}} \left[\text{loss}(\hat{\pi},\check{\pi}) + \theta \cdot \sum_{i=1}^{n} \left(\phi_i(x,\check{\pi}_i) - \phi_i(x,\pi_i)\right)\right] \tag{4.8}$$

$$= \min_{\theta} \mathbb{E}_{(x,\pi)\sim\tilde{P}} \min_{\hat{P}(\hat{\pi}|x)} \max_{\check{P}(\check{\pi}|x)} \mathbb{E}_{\hat{\pi}|x\sim\hat{P};\check{\pi}|x\sim\check{P}} \left[\text{loss}(\hat{\pi},\check{\pi}) + \theta \cdot \sum_{i=1}^{n} \left(\phi_i(x,\check{\pi}_i) - \phi_i(x,\pi_i)\right)\right]. \tag{4.9}$$

The transformation steps above follow the similar transformations in the multiclass classification case, i.e. in the proof of Theorem 2.1.

We use the Hamming distance, $\text{loss}(\hat{\pi},\check{\pi}) = \sum_{i=1}^{n} 1(\hat{\pi}_i \neq \check{\pi}_i)$, as the loss function. Table IX shows the payoff matrix for the game of size $n = 3$ with 3! actions (permutations) for the predictor player $\hat{\pi}$ and for the adversarial approximation player $\check{\pi}$. Here, we define the difference

between the Lagrangian potential of the adversary's action and the ground truth permutation as $\delta_{\check{\pi}} = \psi(\check{\pi}) - \psi(\pi) = \theta \cdot \sum_{i=1}^{n} \left(\phi_i(x, \check{\pi}_i) - \phi_i(x, \pi_i)\right)$.

TABLE IX. Augmented Hamming loss matrix for n=3 permutations.

	$\check{\pi} = 123$	$\check{\pi} = 132$	$\check{\pi} = 213$	$\check{\pi} = 231$	$\check{\pi} = 312$	$\check{\pi} = 321$
$\hat{\pi}=123$	$0 + \delta_{123}$	$2 + \delta_{132}$	$2 + \delta_{213}$	$3 + \delta_{231}$	$3 + \delta_{312}$	$2 + \delta_{321}$
$\hat{\pi}=132$	$2 + \delta_{123}$	$0 + \delta_{132}$	$3 + \delta_{213}$	$2 + \delta_{231}$	$2 + \delta_{312}$	$3 + \delta_{321}$
$\hat{\pi}=213$	$2 + \delta_{123}$	$3 + \delta_{132}$	$0 + \delta_{213}$	$2 + \delta_{231}$	$2 + \delta_{312}$	$3 + \delta_{321}$
$\hat{\pi}=231$	$3 + \delta_{123}$	$2 + \delta_{132}$	$2 + \delta_{213}$	$0 + \delta_{231}$	$3 + \delta_{312}$	$2 + \delta_{321}$
$\hat{\pi}=312$	$3 + \delta_{123}$	$2 + \delta_{132}$	$2 + \delta_{213}$	$3 + \delta_{231}$	$0 + \delta_{312}$	$2 + \delta_{321}$
$\hat{\pi}=321$	$2 + \delta_{123}$	$3 + \delta_{132}$	$3 + \delta_{213}$	$2 + \delta_{231}$	$2 + \delta_{312}$	$0 + \delta_{321}$

Unfortunately, the number of permutations, π, grows factorially ($\mathcal{O}(n!)$) with the number of elements being matched (n). This makes explicit construction of the Lagrangian minimax game intractable for even modestly-sized problems.

4.3.2 Marginal Distribution Formulation

Our approach avoids the need of computing the factorially many permutations in solving the adversarial bipartite matching game by leveraging the key insight that all quantities of interest for evaluating the loss and satisfying the constraints depend only on marginal probabilities

of the permutation's value assignments. Based on this, we employ a marginal distribution decomposition of the game.

We begin this reformulation by first defining a matrix representation of permutation π as $\mathbf{Y}(\pi) \in \mathbb{R}^{n \times n}$ (or simply $\mathbf{Y}$) where the value of its cell $Y_{i,j}$ is 1 when $\pi_i = j$ and 0 otherwise. To be a valid complete bipartite matching or permutation, each column and row of $\mathbf{Y}$ can only have one entry of 1. For each feature function $\phi_i^{(k)}(x, \pi_i)$, we also denote its matrix representation as $\mathbf{X}_k$ whose (i, j)-th cell represents the k-th entry of $\phi_i(x, j)$. For a given distribution of permutations, $P(\pi)$, we denote the marginal probabilities of matching i with j as $p_{i,j} \triangleq P(\pi_i = j)$. We let $\mathbf{P} = \sum_\pi P(\pi)\mathbf{Y}(\pi)$ be the predictor's marginal probability matrix where its (i, j) cell represents $\hat{P}(\hat{\pi}_i = j)$, and similarly let $\mathbf{Q}$ be the adversary's marginal probability matrix (based on $\check{P}$), as shown in Table X.

TABLE X. Doubly stochastic matrices $\mathbf{P}$ and $\mathbf{Q}$ for the marginal decompositions of each player's mixture of permutations.

	1	2	3
$\hat{\pi}_1$	$p_{1,1}$	$p_{1,2}$	$p_{1,3}$
$\hat{\pi}_2$	$p_{2,1}$	$p_{2,2}$	$p_{2,3}$
$\hat{\pi}_3$	$p_{3,1}$	$p_{3,2}$	$p_{3,3}$

	1	2	3
$\check{\pi}_1$	$q_{1,1}$	$q_{1,2}$	$q_{1,3}$
$\check{\pi}_2$	$q_{2,1}$	$q_{2,2}$	$q_{2,3}$
$\check{\pi}_3$	$q_{3,1}$	$q_{3,2}$	$q_{3,3}$

The Birkhoff–von Neumann theorem (Birkhoff, 1946; Von Neumann, 1953) states that the convex hull of the set of $n \times n$ permutation matrices forms a convex polytope in $\mathbb{R}^{n^2}$ (known as

the Birkhoff polytope B_n) in which points are doubly stochastic matrices, i.e., the $n \times n$ matrices with non-negative elements where each row and column must sum to one. This implies that both marginal probability matrices $\mathbf{P}$ and $\mathbf{Q}$ are doubly stochastic matrices. In contrast to the space of distributions over permutation of n objects, which grows factorially ($\mathcal{O}(n!)$ with $n! - 1$ free parameters), the size of this marginal matrices grows only quadratically ($\mathcal{O}(n^2)$ with $n^2 - 2n$ free parameters). This provides a significant benefit in terms of the optimization.

Starting with the minimax over $\hat{P}(\hat{\pi})$ and $\check{P}(\check{\pi})$ in the permutation mixture formulation, and using the matrix notation above, we rewrite Equation (4.3) as a minimax over marginal probability matrices $\mathbf{P}$ and $\mathbf{Q}$ with additional constraints that both $\mathbf{P}$ and $\mathbf{Q}$ are doubly-stochastic matrices, i.e., $\mathbf{P} \geq \mathbf{0}$ (elementwise), $\mathbf{Q} \geq \mathbf{0}$, $\mathbf{P1} = \mathbf{P}^\top \mathbf{1} = \mathbf{Q1} = \mathbf{Q}^\top \mathbf{1} = \mathbf{1}$ where $\mathbf{1} = (1, \ldots, 1)^\top$). That is:

$$\min_{\theta} \mathbb{E}_{\mathbf{X},\mathbf{Y} \sim \tilde{P}} \min_{\mathbf{P} \geq \mathbf{0}} \max_{\mathbf{Q} \geq \mathbf{0}} [n - \langle \mathbf{P}, \mathbf{Q} \rangle + \langle \mathbf{Q} - \mathbf{Y}, \textstyle\sum_k \theta_k \mathbf{X}_k \rangle] \tag{4.10}$$

$$\text{subject to: } \mathbf{P1} = \mathbf{P}^\top \mathbf{1} = \mathbf{Q1} = \mathbf{Q}^\top \mathbf{1} = \mathbf{1},$$

where $\langle \cdot, \cdot \rangle$ denotes the Frobenius inner product between two matrices, i.e., $\langle A, B \rangle = \sum_{i,j} A_{i,j} B_{i,j}$.

4.3.2.1 Optimization

We reduce the computational costs of the optimization in Equation (4.10) by focusing on optimizing the adversary's marginal probability $\mathbf{Q}$. By strong duality, we then push the maximization over $\mathbf{Q}$ in the formulation above to the outermost level of Eq. Equation (4.10). Note that the objective above is a non-smooth function (i.e., piece-wise linear). For the purpose

of smoothing the objective, we add a small amount of strongly convex prox-functions to both $\mathbf{P}$ and $\mathbf{Q}$. We also add a regularization penalty to the parameter θ to improve the generalizability of our model. We unfold Equation (4.10) by replacing the empirical expectation with an average over all training examples, resulting in the following optimization:

$$\max_{\mathbf{Q}\geq\mathbf{0}} \min_{\theta} \frac{1}{m}\sum_{i=1}^{m} \min_{\mathbf{P}_i\geq\mathbf{0}} \Big[\langle \mathbf{Q}_i - \mathbf{Y}_i, \textstyle\sum_k \theta_k \mathbf{X}_{i,k}\rangle - \langle \mathbf{P}_i, \mathbf{Q}_i\rangle + \frac{\mu}{2}\|\mathbf{P}_i\|_F^2 - \frac{\mu}{2}\|\mathbf{Q}_i\|_F^2 \Big] + \frac{\lambda}{2}\|\theta\|_2^2 \quad (4.11)$$

$$\text{subject to: } \mathbf{P}_i\mathbf{1} = \mathbf{P}_i^\top\mathbf{1} = \mathbf{Q}_i\mathbf{1} = \mathbf{Q}_i^\top\mathbf{1} = \mathbf{1}, \quad \forall i,$$

where m is the number of bipartite matching problems in the training set, λ is the regularization penalty parameter, μ is the smoothing penalty parameter, and $\|A\|_F$ denotes the Frobenius norm of matrix A. The subscript i in $\mathbf{P}_i, \mathbf{Q}_i, \mathbf{X}_i$, and $\mathbf{Y}_i$ refers to the i-th example in the training set.

In the formulation above, given a fixed $\mathbf{Q}$, the inner minimization over θ and $\mathbf{P}$ can then be solved separately. The optimal θ in the inner minimization admits a closed-form solution, in which the k-th element of θ^* is:

$$\theta_k^* = -\frac{1}{\lambda m}\sum_{i=1}^{m} \langle \mathbf{Q}_i - \mathbf{Y}_i, \mathbf{X}_{i,k}\rangle \,. \quad (4.12)$$

The inner minimization over $\mathbf{P}$ can be solved independently for each training example. Given the adversary's marginal probability matrix $\mathbf{Q}_i$ for the i-th example, the optimal $\mathbf{P}_i$ can be formulated as:

$$\mathbf{P}_i^* = \underset{\{\mathbf{P}_i \geq \mathbf{0} | \mathbf{P}_i \mathbf{1} = \mathbf{P}_i^\top \mathbf{1} = \mathbf{1}\}}{\operatorname{argmin}} \frac{\mu}{2} \|\mathbf{P}_i\|_F^2 - \langle \mathbf{P}_i, \mathbf{Q}_i \rangle \tag{4.13}$$

$$= \underset{\{\mathbf{P}_i \geq \mathbf{0} | \mathbf{P}_i \mathbf{1} = \mathbf{P}_i^\top \mathbf{1} = \mathbf{1}\}}{\operatorname{argmin}} \|\mathbf{P}_i - \tfrac{1}{\mu} \mathbf{Q}_i\|_F^2. \tag{4.14}$$

We can interpret this minimization as projecting the matrix $\frac{1}{\mu}\mathbf{Q}_i$ to the set of doubly-stochastic matrices. We will discuss our projection technique in the upcoming subsection.

For solving the outer optimization over $\mathbf{Q}$ with the doubly-stochastic constraints, we employ a projected Quasi-Newton algorithm (Schmidt et al., 2009). Each iteration of the algorithm optimizes the quadratic approximation of the objective function (using limited-memory Quasi-Newton) over the the convex set. In each update step, a projection to the set of doubly-stochastic matrices is needed, akin to the inner minimization of $\mathbf{P}$ in Equation (4.14).

The optimization above provides the adversary's optimal marginal probability $\mathbf{Q}^*$. To achieve our learning goal, we recover θ^* using Equation (4.12) computed over the optimal $\mathbf{Q}^*$. We use the θ^* that our model learns from this optimization to construct a weighted bipartite graph for making predictions for test examples.

4.3.2.2 Doubly-Stochastic Matrix Projection

The projection from an arbitrary matrix $\mathbf{R}$ to the set of doubly-stochastic matrices can be formulated as:

$$\min_{\mathbf{P}\geq\mathbf{0}} \|\mathbf{P}-\mathbf{R}\|_F^2, \qquad \text{subject to: } \mathbf{P1}=\mathbf{P}^\top\mathbf{1}=\mathbf{1}. \tag{4.15}$$

We employ the alternating direction method of multipliers (ADMM) technique (Douglas and Rachford, 1956; Glowinski and Marroco, 1975; Boyd et al., 2011) to solve the optimization problem above. We divide the doubly-stochastic matrix constraint into two sets of constraints $C_1 : \mathbf{P1}=\mathbf{1}$ and $\mathbf{P}\geq\mathbf{0}$, and $C_2 : \mathbf{P}^\top\mathbf{1}=\mathbf{1}$ and $\mathbf{P}\geq\mathbf{0}$. Using this construction, we convert the optimization above into ADMM form as follows:

$$\begin{aligned} &\min_{\mathbf{P},\mathbf{S}} \tfrac{1}{2}\|\mathbf{P}-\mathbf{R}\|_F^2 + \tfrac{1}{2}\|\mathbf{S}-\mathbf{R}\|_F^2 + \mathrm{I}_{C_1}(\mathbf{P}) + \mathrm{I}_{C_2}(\mathbf{S}) \\ &\quad \text{subject to: } \mathbf{P}-\mathbf{S}=0. \end{aligned} \tag{4.16}$$

The augmented Lagrangian for this optimization is:

$$\mathcal{L}_\rho(\mathbf{P},\mathbf{S},\mathbf{W}) = \tfrac{1}{2}\|\mathbf{P}-\mathbf{R}\|_F^2 + \tfrac{1}{2}\|\mathbf{S}-\mathbf{R}\|_F^2 + \mathrm{I}_{C_1}(\mathbf{P}) + \mathrm{I}_{C_2}(\mathbf{S}) + \tfrac{\rho}{2}\|\mathbf{P}-\mathbf{S}+\mathbf{W}\|_F^2, \tag{4.17}$$

where ρ is the ADMM penalty parameter and $\mathbf{W}$ is the scaled dual variable. From the augmented Lagrangian, we compute the update for $\mathbf{P}$ as:

$$\mathbf{P}^{t+1} = \underset{\mathbf{P}}{\operatorname{argmin}}\, \mathcal{L}_\rho(\mathbf{P}, \mathbf{S}^t, \mathbf{W}^t) \tag{4.18}$$

$$= \underset{\{\mathbf{P}\geq\mathbf{0}|\mathbf{P1}=\mathbf{1}\}}{\operatorname{argmin}}\, \tfrac{1}{2}\|\mathbf{P}-\mathbf{R}\|_F^2 + \tfrac{\rho}{2}\|\mathbf{P}-\mathbf{S}^t+\mathbf{W}^t\|_F^2 \tag{4.19}$$

$$= \underset{\{\mathbf{P}\geq\mathbf{0}|\mathbf{P1}=\mathbf{1}\}}{\operatorname{argmin}}\, \|\mathbf{P} - \tfrac{1}{1+\rho}\left(\mathbf{R}+\rho\left(\mathbf{S}^t-\mathbf{W}^t\right)\right)\|_F^2. \tag{4.20}$$

The minimization above can be interpreted as a projection to the set $\{\mathbf{P}\geq 0|\mathbf{P1}=\mathbf{1}\}$ which can be realized by projecting to the probability simplex independently for each row of the matrix $\frac{1}{1+\rho}\left(\mathbf{R}+\rho\left(\mathbf{S}^t-\mathbf{W}^t\right)\right)$. Similarly, the ADMM update for $\mathbf{S}$ can also be formulated as a column-wise probability simplex projection. The technique for projecting a point to the probability simplex has been studied previously, e.g., by (Duchi et al., 2008). Therefore, our ADMM algorithm consists of the following updates:

$$\mathbf{P}^{t+1} = \text{Proj}_{C_1}\left(\tfrac{1}{1+\rho}\left(\mathbf{R}+\rho\left(\mathbf{S}^t-\mathbf{W}^t\right)\right)\right) \tag{4.21}$$

$$\mathbf{S}^{t+1} = \text{Proj}_{C_2}\left(\tfrac{1}{1+\rho}\left(\mathbf{R}+\rho\left(\mathbf{P}^{t+1}+\mathbf{W}^t\right)\right)\right) \tag{4.22}$$

$$\mathbf{W}^{t+1} = \mathbf{W}^t + \mathbf{P}^{t+1} - \mathbf{S}^{t+1}. \tag{4.23}$$

We run this series of updates until the stopping conditions are met. Our stopping conditions are based on the primal and dual residual optimality as described in (Boyd et al., 2011). In our overall algorithm, this ADMM projection algorithm is used both in the projected Quasi-Newton

algorithm for optimizing $\mathbf{Q}$ (Equation (4.11)) and in the inner optimization for minimizing $\mathbf{P}_i$ (Equation (4.14)).

4.3.2.3 Convergence Property

The convergence rate of ADMM is $\mathcal{O}(\log \frac{1}{\epsilon})$ thanks to the strong convexity of the objective (Deng and Yin, 2016). Each step inside ADMM is simply a projection to a simplex, hence costing $\tilde{\mathcal{O}}(n)$ computations (Duchi et al., 2008).

In terms of optimization on $\mathbf{Q}$, since no explicit rates of convergence are available for the projected Quasi-Newton algorithm (Schmidt et al., 2009) that finely characterize the dependency on the condition numbers, we simply illustrate the $\sqrt{L/\mu}\log\frac{1}{\epsilon}$ rate using Nesterov's accelerated gradient algorithm (Nesterov, 2003), where L is the Lipschitz continuous constant of the gradient. In our case, $L = \frac{1}{m^2\lambda}\sum_k \sum_{i=1}^m \|\mathbf{X}_{i,k}\|_F^2 + 1/\mu$.

Comparison with Structured SVM (SSVM). Conventional SSVMs for learning bipartite matchings have only $\mathcal{O}(1/\epsilon)$ rates due to the lack of smoothness (Joachims et al., 2009; Teo et al., 2010). If smoothing is added, then similar linear convergence rates can be achieved with similar condition numbers. However, it is noteworthy that at each iteration we need to apply ADMM to solve a projection problem to the doubly stochastic matrix set (Equation (4.15)), while SSVMs (without smoothing) solves a matching problem with the Hungarian algorithm, incurring $\mathcal{O}(n^3)$ time.

4.3.3 Fisher Consistency Analysis

Despite its apparent differences from standard empirical risk minimization (ERM), adversarial loss minimization (Equation (4.3)) can be equivalently recast as an ERM:

$$\min_{\theta} \mathbb{E}_{\substack{x\sim P \\ \pi|x\sim\tilde{P}}} \left[AL^{\text{perm}}_{f_\theta}(x,\pi) \right] \text{ where } AL^{\text{perm}}_{f_\theta}(x,\pi) \triangleq \tag{4.24}$$
$$\min_{\hat{P}(\hat{\pi}|x)} \max_{\check{P}(\check{\pi}|x)} \mathbb{E}_{\substack{\hat{\pi}|x\sim\hat{P} \\ \check{\pi}|x\sim\check{P}}} \left[\text{loss}(\hat{\pi},\check{\pi}) + f_\theta(x,\check{\pi}) - f_\theta(x,\pi) \right]$$

and $f_\theta(x,\pi) = \theta \cdot \sum_{i=1}^{n} \phi(x,\pi_i)$ is the Lagrangian potential function. Here we consider f_θ as the linear discriminant function for a proposed permutation π, using parameter value θ. $AL^{\text{perm}}_{f_\theta}(x,\pi)$ is then the surrogate loss for input x and permutation π.

As described in Section 4.2.2, Fisher consistency is an important property for a surrogate loss L. It requires that under the true distribution $P(x,\pi)$, the hypothesis that minimizes L is Bayes optimal (Tewari and Bartlett, 2007; Liu, 2007). In our setting, the Fisher consistency of AL^{perm}_{f} can be written as:

$$f^* \in \mathcal{F}^* \triangleq \operatorname*{argmin}_{f} \mathbb{E}_{\pi|x\sim P} \left[AL^{\text{perm}}_{f}(x,\pi) \right] \tag{4.25}$$
$$\Rightarrow \operatorname*{argmax}_{\pi} f^*(x,\pi) \subseteq \Pi^{\diamond} \triangleq \operatorname*{argmin}_{\pi} \mathbb{E}_{\bar{\pi}|x\sim P}[\text{loss}(\pi,\bar{\pi})].$$

Note that in Equation (4.25) we allow f to be optimized over the set of all measurable functions on the input space (x,π). In our formulation, we have restricted f to be additively decomposable over individual elements of permutation, $f(x,\pi) = \sum_i g_i(x,\pi_i)$. In the sequel, we

will show that the condition in Equation (4.25) also holds for this restricted set provided that g is allowed to be optimized over the set of all measurable functions on the space of individual input (x, π_i).

Theorem 4.1. *Suppose we have a loss metric that satisfy the natural requirement of loss*$(\pi, \pi) <$ *loss*$(\bar{\pi}, \pi)$ *for all* $\bar{\pi} \neq \pi$. *Then the adversarial permutation loss* AL_f^{perm} *is Fisher consistent if* f *is over all measurable functions on the input space* (x, π).

Proof. Given that f is optimized over all measurable functions on the input space (x, π), the minimization in Equation (4.25) is equivalent with the case of multiclass classification where the number of class is $n!$, comprises of all possible permutations. Based on our analysis in Section 2.6.1, AL_f^{perm} is Fisher consistent in this case. □

Theorem 4.2. *Suppose the loss is Hamming loss, and the potential function* $f(x, \pi)$ *decomposes additively by* $\sum_i g_i(x, \pi_i)$. *Then, the adversarial permutation loss* AL_f^{perm} *is Fisher consistent provided that* g_i *is allowed to be optimized over the set of all measurable functions on the space of individual inputs* (x, π_i).

Proof. Simply choose g_i such that for each sample x in the population, $g_i(x, \pi_i) = -(\pi_i \neq \pi_i^{\diamond})$. This renders the loss reflective property under the Hamming loss. □

4.4 Experimental Evaluation

To evaluate our approach, we apply our adversarial bipartite matching model to video tracking tasks using public benchmark datasets (Leal-Taixé et al., 2015). In this problem, we are given a set of images (video frames) and a list of objects in each image. We are also given the

correspondence matching between objects in frame t and objects in frame $t+1$. Figure 14 shows an example of the problem setup. It is important to note that the number of objects are not the same in every frame. Some of the objects may enter, leave, or remain in the consecutive frames. To handle this issue, we setup our experiment as follows. Let k_t be the number of objects in frame t and k^* be the maximum number of objects a frame can have, i.e., $k^* = \max_{t \in T} k_t$. Starting from k^* nodes to represent the objects, we add k^* more nodes as "invisible" nodes to allow new objects to enter and existing objects to leave. As a result, the total number of nodes in each frame doubles to $n = 2k^*$.

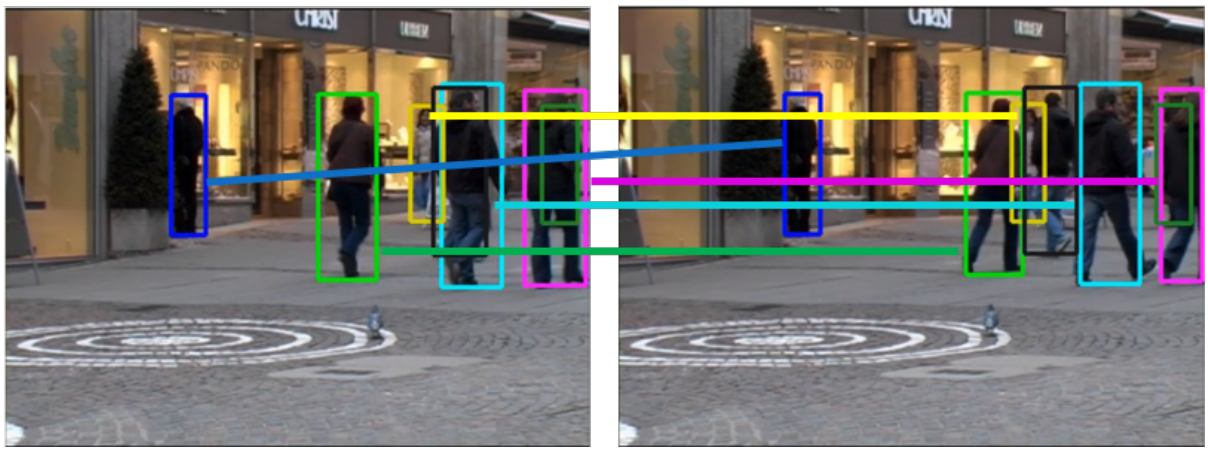


Figure 14. An example of bipartite matching in video tracking.

4.4.1 Feature Representation

We define the features for pairs of bounding boxes (i.e., $\phi_i(x, j)$ for pairing bounding box i with bounding box j) in two consecutive video frames so that we can compute the associative feature vectors, $\phi(x, \pi) = \sum_{i=1}^{n} \phi_i(x, \pi_i)$, for each possible matching π. To define the feature vector $\phi_i(\cdot, \cdot)$, we follow the feature representation reported by (Kim et al., 2012) using six different types of features: (1) intersection over union (IoU) overlap ratio between bounding boxes, (2) Euclidean distance between object centers, (3) 21 color histogram distance features (RGB) from the Bhattacharyaa distance, (4) 21 local binary pattern (LBP) features, (5) optical flow (motion) between bounding boxes, and (6) three indicator variables (for *entering*, *leaving*, and *staying invisible)*.

4.4.2 Experimental Setup

We compare our approach with the Structured SVM (SSVM) model (Taskar et al., 2005a; Tsochantaridis et al., 2005) implemented based on (Kim et al., 2012) using `SVM-Struct` (Joachims, 2008; Vedaldi, 2011). We implement our optimization algorithm using `minConf` (Schmidt, 2008) for performing projected Quasi-Newton optimization.

We consider two different groups of datasets in our experiment: TUD datasets and ETH datasets. Each dataset contains different numbers of elements (i.e., the number of pedestrian bounding box in the frame plus the number of extra nodes to indicate entering or leaving) and different numbers of examples (i.e., pairs of two consecutive frames that we want to match). Table XI contains the detailed information about the datasets.

TABLE XI. Dataset properties

Dataset	# Elements	# Examples
TUD-Campus	12	70
TUD-Stadtmitte	16	178
ETH-Sunnyday	18	353
ETH-Bahnhof	34	999
ETH-Pedcross2	30	836

To avoid having test examples that are too similar with the training set, we train the models on one dataset and test the model on another dataset that has similar characteristics. In particular, we perform evaluations for every pair of datasets in TUD and ETH collections. This results in eight pairs of training/test datasets, as shown in Table XII.

To tune the regularization parameter (λ in adversarial matching, and C in SSVM), we perform 5-fold cross validation based on the training dataset only. The resulting best regularization parameter is used to train the model over all training examples to obtain parameters θ, which we then use to predict the matching for the testing data. For both SSVM and our method, the prediction is done by finding the bipartite matching that maximizes the potential value, i.e., $\text{argmax}_{\mathbf{Y}} \langle \mathbf{Y}, \sum_k \theta_k \mathbf{X}_k \rangle$ which can be solved using the Hungarian algorithm.

4.4.3 Results

We report the average accuracy, which in this case is defined as (1 − the average Hamming loss) over all examples in the testing dataset. Table XII shows the mean and the standard deviation of our metric across different dataset pairs. We highlight (using bold font) the cases

in which our result is better with statistical significance (under Wilcoxon signed-rank test with $\alpha = 0.05$) in Table XII. Compared with SSVM, our proposed adversarial matching outperforms SSVM in all pairs of datasets—with statistical significance on all six pairs of the ETH datasets and slightly better than SSVM on the TUD datasets. This suggests that our adversarial bipartite matching model benefits from its Fisher consistency property.

TABLE XII. The mean and standard deviation (in parenthesis) of the average accuracy (1 - the average Hamming loss) for the adversarial bipartite matching model compared with the structured-SVM.

Training / Testing	Adv. Marginal	SSVM
Campus / Stadtmitte	0.662 (0.08)	0.662 (0.08)
Stadtmitte / Campus	0.667 (0.11)	0.660 (0.12)
Bahnhof / Sunnyday	**0.754** (0.10)	0.729 (0.15)
Pedcross2 / Sunnyday	**0.750** (0.10)	0.736 (0.13)
Sunnyday/ Bahnhof	**0.751** (0.18)	0.739 (0.20)
Pedcross2 / Bahnhof	**0.763** (0.16)	0.731 (0.21)
Bahnhof / Pedcross2	**0.714** (0.16)	0.701 (0.18)
Sunnyday / Pedcross2	**0.712** (0.17)	0.700 (0.18)

In terms of the running time, Table XIII shows that our marginal formulation is relatively fast. It only takes a few seconds to train until convergence in the case of 50 examples, with the number of elements varied up to 34. The running time grows roughly quadratically in the number of elements, which is natural since the size of the marginal probability matrices $\mathbf{P}$ and

TABLE XIII. Running time (in seconds) of the model for various number of elements n with fixed number of samples ($m = 50$)

Dataset	# Elements	Adv. Marginal	SSVM
Campus	12	1.96	0.22
Stadtmitte	16	2.46	0.25
Sunnyday	18	2.75	0.15
Pedcross2	30	8.18	0.26
Bahnhof	34	9.79	0.31

Q also grow quadratically in the number of elements. This shows that our approach is much more efficient than the CRF approach, which has a running time that is impractical even for small problems with 20 elements. The training time of SSVM is faster than the adversarial methods due to two different factors: (1) the inner optimization of SSVM can be solved using a single execution of the Hungarian algorithm compared with the inner optimization of adversarial method which requires ADMM optimization for projection to doubly stochastic matrix set; (2) different tools for implementation, i.e., C++ for SSVM and MATLAB for our method, which benefits the running time of SSVM.

4.5 Conclusions and Future Works

We have presented an adversarial approach for learning bipartite matchings that is not only computationally efficient to employ but also provides Fisher consistency guarantees. We showed that these theoretical advantages translate into better empirical performance for our

model compared with previous approaches. Our future work will explore matching problems with different loss functions and other graphical structures.

CHAPTER 5

CONCLUSIONS AND FUTURE DIRECTIONS

5.1 Conclusions

This thesis proposed a family of learning algorithms that combine the strengths of probabilistic learning approach in its statistical guarantee of Fisher consistency, with the strengths of large-margin approach in its flexibility to align with custom performance/loss metrics and its computational efficiency. This also avoids the main drawbacks of the probabilistic and large-margin approaches. Our proposed methods find a predictor that maximize the performance metric (or minimize the loss metric) in the worst case given the statistical summaries of the empirical distributions. We presented the formulations, theoretical properties, optimization algorithms, and practical benefits of our learning algorithms in two different areas, general multiclass classification and structured prediction.

In the general multiclass classification problems, our formulation can be viewed as surrogate losses over the desired loss metrics. We presented efficient algorithms to compute the surrogate loss and proved their Fisher consistency property. We designed efficient learning algorithms and also a way to incorporate richer features via the kernel trick. We then demonstrated the benefit of our approach compared with the state-of-the-art piece-wise linear surrogate losses which includes many different forms of multiclass SVMs and their extension to many general multiclass classification problems.

In the structured prediction area, we focused on two important problems: graphical models and bipartite matching. Our proposed models provide the flexibility of incorporating custom loss/performance metric in their learning process, the statistical guarantee of Fisher consistency, and the computational efficiency by optimizing over the marginal distributions of each model. This benefits many application problems where aligning the learning algorithms with custom loss metrics is desirable, for example in many natural language and computer vision tasks.

5.2 Future Directions

There are several possible future directions that can be explored by leveraging the proposed methods in this thesis. Among those directions are: fairness in machine learning, structured performance/loss metrics, and more general structures in graphical models.

Fairness in Machine Learning

As there are growing numbers of machine learning applications in automated decisions that can impact people's lives, the need to build learning algorithms that ensure fairness among the users is also growing. This requires machine learning predictors to produce fair predictions (Hardt et al., 2016; Zafar et al., 2017; Dwork et al., 2012). From the perspective of our adversarial formulation, we currently only enforce constraints on the adversary to match the statistics of the data. Fairness constraints can be added to the model by also constraining the predictor to output fair prediction.

Structured Performance/Loss Metrics

For many standard classification problems, even though the problem itself is not a structured prediction problem, i.e., only predict single variable y for given $\mathbf{x}$, the performance metric in

which the prediction is evaluated has some structure. Some examples of the desired structured performance metrics are precision, recall, F_β-score, Jaccard score, and ROC-area. A possible research direction can be explored to develop a plug-in classifier for these performance metrics that enjoys Fisher consistency based on the robust adversarial formulation. This will provide a Fisher consistent alternative to the SVM^{perf} algorithm (Joachims, 2005). Previous works (Wang et al., 2015; Shi et al., 2017) have investigated the problem for certain performance metrics, but these approaches require significant modifications (in some cases, rewriting the whole algorithm) when different metrics are used.

Structured Prediction and Graphical Models

The extension of the Adversarial Graphical Models (AGM) to more complex graphical structures and more general performance/loss metrics is also an interesting future direction. While this thesis focused on graphical structures that admit tractable optimization, the case where exact learning and inference are intractable need to be further investigated. This case includes more complex lattice-based graphical structures which are popular in computer vision applications, and more generally, graphical structures with loops.

APPENDIX

COPYRIGHT POLICIES

A.1 Copyright Policy of Neural Information Processing Systems

The Neural Information Processing Systems conference's acronym changed from NIPS to NeurIPS in 2018.

> All NIPS/NeurIPS authors retain copyright of their work. You will need to sign a nonexclusive license giving the NIPS/NeurIPS foundation permission to publish the work. Ultimately, however, you can do whatever you like with the content, including having the paper as a chapter of your thesis.

A.2 Copyright Policy of the International Conference on Machine Learning

The International Conference on Machine Learning (ICML) conference's proceeding is published by the Proceedings of Machine Learning Research (PMLR).

> The Proceedings of Machine Learning Research (formerly JMLR Workshop and Conference Proceedings) is a series aimed specifically at publishing machine learning research presented at workshops and conferences. Each volume is separately titled and associated with a particular workshop or conference and will be published online on the PMLR web site. Authors will retain copyright and individual volume editors are free to make additional hardcopy publishing arrangements (see for example the

APPENDIX (Continued)

Challenges in Machine Learning series which includes free PDFs and low cost hard copies), but PMLR will not produce hardcopies of these volumes.

A.3 Copyright Policy of ArXiv

Our paper (Fathony et al., 2018c) is published in arXiv with the "non-exclusive and irrevocable license".

ArXiv does not ask that copyright be transferred. However, we require sufficient rights to allow us to distribute submitted articles in perpetuity. In order to submit an article to arXiv, the submitter must either:

- grant arXiv.org a non-exclusive and irrevocable license to distribute the article, and certify that he/she has the right to grant this license;
- certify that the work is available under one of the following Creative Commons licenses and that he/she has the right to assign this license:
  - Creative Commons Attribution license (CC BY 4.0)
  - Creative Commons Attribution-ShareAlike license (CC BY-SA 4.0)
  - Creative Commons Attribution-Noncommercial-ShareAlike license (CC BY-NC-SA 4.0);
- or dedicate the work to the public domain by associating the Creative Commons Public Domain Dedication (CC0 1.0) with the submission.

In the most common case, authors have the right to grant these licenses because they hold copyright in their own work.

CITED LITERATURE

Albertsson, K., Altoe, P., Anderson, D., Andrews, M., Espinosa, J. P. A., Aurisano, A., Basara, L., Bevan, A., Bhimji, W., Bonacorsi, D., et al.: Machine learning in high energy physics community white paper. In Journal of Physics: Conference Series, volume 1085, page 022008. IOP Publishing, 2018.

Amini, M. R., Truong, T. V., and Goutte, C.: A boosting algorithm for learning bipartite ranking functions with partially labeled data. In Proceedings of the International ACM SIGIR Conference, pages 99–106. ACM, 2008.

Andréasson, N., Evgrafov, A., Patriksson, M., Gustavsson, E., and Önnheim, M.: An introduction to continuous optimization: foundations and fundamental algorithms, volume 28. Studentlitteratur Lund, 2005.

Asif, K., Xing, W., Behpour, S., and Ziebart, B. D.: Adversarial cost-sensitive classification. In Proceedings of the Conference on Uncertainty in Artificial Intelligence, pages 92–101, 2015.

Baccianella, S., Esuli, A., and Sebastiani, F.: Evaluation measures for ordinal regression. In 2009 Ninth International Conference on Intelligent Systems Design and Applications, pages 283–287. IEEE, 2009.

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

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

Baum, L. E. and Petrie, T.: Statistical inference for probabilistic functions of finite state markov chains. The Annals of Mathematical Statistics, 37(6):1554–1563, 1966.

Belongie, S., Malik, J., and Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(4):509–522, 2002.

Bertsekas, D. P.: Control of Uncertain Systems with a Set-Membership Description of Uncertainty. Doctoral dissertation, MIT, 1971.

Binder, A., Müller, K.-R., and Kawanabe, M.: On taxonomies for multi-class image categorization. International Journal of Computer Vision, 99(3):281–301, 2012.

Birkhoff, G.: Three observations on linear algebra. Univ. Nac. Tacuman, Rev. Ser. A, 5:147–151, 1946.

Blunsom, P. and Cohn, T.: Discriminative word alignment with conditional random fields. In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, pages 65–72. Association for Computational Linguistics, 2006.

Boser, B. E., Guyon, I. M., and Vapnik, V. N.: A training algorithm for optimal margin classifiers. In Proceedings of the Workshop on Computational Learning Theory, pages 144–152. ACM, 1992.

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends® in Machine Learning, 3(1):1–122, 2011.

Boyd, S. and Vandenberghe, L.: Convex optimization. Cambridge University Press, 2004.

Boyd, S., Xiao, L., Mutapcic, A., and Mattingley, J.: Notes on decomposition methods. Notes for EE364B, 2008.

Carreras, X. and Màrquez, L.: Introduction to the conll-2005 shared task: Semantic role labeling. In Proceedings of the Ninth Conference on Computational Natural Language Learning, CONLL '05, pages 152–164. Association for Computational Linguistics, 2005.

Chan, Y. S. and Ng, H. T.: Maxsim: A maximum similarity metric for machine translation evaluation. Proceedings of ACL-08: HLT, pages 55–62, 2008.

Chen, D., Fisch, A., Weston, J., and Bordes, A.: Reading wikipedia to answer open-domain questions. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pages 1870–1879, 2017.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In The European Conference on Computer Vision (ECCV), September 2018.

Chen, R. and Paschalidis, I. C.: A robust learning approach for regression models based on distributionally robust optimization. Journal of Machine Learning Research, 19(13):1–48, 2018.

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

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

Ciliberto, C., Rosasco, L., and Rudi, A.: A consistent regularization approach for structured prediction. In Advances in Neural Information Processing Systems, pages 4412–4420, 2016.

Cohn, T. and Blunsom, P.: Semantic role labelling with tree conditional random fields. In Proceedings of the Ninth Conference on Computational Natural Language Learning, pages 169–172. Association for Computational Linguistics, 2005.

Cooper, G. F.: The computational complexity of probabilistic inference using Bayesian belief networks. Artificial intelligence, 42(2-3):393–405, 1990.

Cortes, C., DeSalvo, G., and Mohri, M.: Boosting with abstention. In Advances in Neural Information Processing Systems, pages 1660–1668, 2016.

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

Cowell, R. G., Dawid, P., Lauritzen, S. L., and Spiegelhalter, D. J.: Probabilistic networks and expert systems: Exact computational methods for Bayesian networks. Springer Science & Business Media, 2006.

Cox, D. R.: The regression analysis of binary sequences. Journal of the Royal Statistical Society. Series B (Methodological), pages 215–242, 1958.

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

Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. In Advances in Neural Information Processing Systems, pages 2292–2300, 2013.

Dantzig, G.: Linear programming and extensions. RAND Corporation, 1963.

Dantzig, G. B.: Programming in a linear structure. Washington, DC, 1948.

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

Dellaert, F., Seitz, S. M., Thorpe, C. E., and Thrun, S.: EM, MCMC, and chain flipping for structure from motion with unknown correspondence. Machine Learning, 50(1-2):45–71, 2003.

Deng, N., Tian, Y., and Zhang, C.: Support vector machines: optimization based theory, algorithms, and extensions. CRC press, 2012.

Deng, W. and Yin, W.: On the global and linear convergence of the generalized alternating direction method of multipliers. Journal of Scientific Computing, 66(3), 2016.

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

Douglas, J. and Rachford, H. H.: On the numerical solution of heat conduction problems in two and three space variables. Transactions of the American Mathematical Society, 82(2):421–439, 1956.

Duchi, J., Shalev-Shwartz, S., Singer, Y., and Chandra, T.: Efficient projections onto the l 1-ball for learning in high dimensions. In Proceedings of the International Conference on Machine Learning, pages 272–279. ACM, 2008.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R.: Fairness through awareness. In Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, pages 214–226. ACM, 2012.

Dwork, C., Kumar, R., Naor, M., and Sivakumar, D.: Rank aggregation methods for the web. In Proceedings of the International Conference on World Wide Web, pages 613–622. ACM, 2001.

Esfahani, P. M. and Kuhn, D.: Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations. Mathematical Programming, 171(1-2):115–166, 2018.

Fathony, R., Asif, K., Liu, A., Bashiri, M. A., Xing, W., Behpour, S., Zhang, X., and Ziebart, B. D.: Consistent robust adversarial prediction for general multiclass classification. arXiv preprint arXiv:1812.07526, 2018c.

Fathony, R., Bashiri, M. A., and Ziebart, B.: Adversarial surrogate losses for ordinal regression. In Advances in Neural Information Processing Systems 30 (NIPS 2017), pages 563–573, 2017.

Fathony, R., Behpour, S., Zhang, X., and Ziebart, B.: Efficient and consistent adversarial bipartite matching. In Proceedings of the 35th International Conference on Machine Learning (ICML 2017), volume 80 of Proceedings of Machine Learning Research, pages 1457–1466, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018a. PMLR.

Fathony, R., Liu, A., Asif, K., and Ziebart, B.: Adversarial multiclass classification: A risk minimization perspective. In Advances in Neural Information Processing Systems 29 (NIPS 2016), pages 559–567, 2016.

Fathony, R., Rezaei, A., Bashiri, M. A., Zhang, X., and Ziebart, B.: Distributionally robust graphical models. In Advances in Neural Information Processing Systems 31 (NeurIPS 2018), pages 8353–8364, 2018b.

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

Glowinski, R. and Marroco, A.: Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires. Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique, 9(R2):41–76, 1975.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y.: Deep learning, volume 1. MIT press Cambridge, 2016.

Grandvalet, Y., Rakotomamonjy, A., Keshet, J., and Canu, S.: Support vector machines with a reject option. In Advances in Neural Information Processing Systems, pages 537–544, 2009.

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

Hardt, M., Price, E., Srebro, N., et al.: Equality of opportunity in supervised learning. In Advances in Neural Information Processing Systems, pages 3315–3323, 2016.

Hashimoto, T., Srivastava, M., Namkoong, H., and Liang, P.: Fairness without demographics in repeated loss minimization. In Proceedings of the 35th International Conference on Machine Learning, volume 80, pages 1929–1938. PMLR, 2018.

Hatori, J., Miyao, Y., and Tsujii, J.: Word sense disambiguation for all words using tree-structured conditional random fields. Coling 2008: Companion Volume: Posters, pages 43–46, 2008.

He, H. and Ma, Y.: Imbalanced learning: foundations, algorithms, and applications. John Wiley & Sons, 2013.

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

Hu, J., Shen, L., and Sun, G.: Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7132–7141, 2018.

Huber, M. and Law, J.: Fast approximation of the permanent for very dense problems. In Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pages 681–689. Society for Industrial and Applied Mathematics, 2008.

Igel, C., Heidrich-Meisner, V., and Glasmachers, T.: Shark. Journal of Machine Learning Research, 9:993–996, 2008.

Joachims, T.: SVM-struct: Support vector machine for complex outputs. `http://www.cs.cornell.edu/People/tj/svm_light/svm_struct.html`, 2008.

Joachims, T., Finley, T., and Yu, C.-N.: Cutting-plane training of structural SVMs. Machine Learning, 77(1):27–59, 2009.

Joachims, T.: A support vector method for multivariate performance measures. In Proceedings of the International Conference on Machine Learning, pages 377–384, 2005.

Karmarkar, N.: 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.

Kim, M. and Pavlovic, V.: Structured output ordinal regression for dynamic facial emotion intensity prediction. In European Conference on Computer Vision, pages 649–662. Springer, 2010.

Kim, S., Kwak, S., Feyereisl, J., and Han, B.: Online multi-target tracking by large margin structured learning. In Asian Conference on Computer Vision, pages 98–111. Springer, 2012.

Kuhn, H. W.: The hungarian method for the assignment problem. Naval Research Logistics, 2(1-2):83–97, 1955.

Lafferty, J., McCallum, A., and Pereira, F. C.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the 18th International Conference on Machine Learning, volume 951, pages 282–289, 2001.

Lample, G., Conneau, A., Ranzato, M., Denoyer, L., and Jégou, H.: Word translation without parallel data. In International Conference on Learning Representations, 2018.

Le, Q. and Smola, A.: Direct optimization of ranking measures. arXiv preprint arXiv:0704.3359, 2007.

Leal-Taixé, L., Milan, A., Reid, I., Roth, S., and Schindler, K.: Motchallenge 2015: Towards a benchmark for multi-target tracking. arXiv preprint arXiv:1504.01942, 2015.

Lee, Y., Lin, Y., and Wahba, G.: 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.

Li, L. and Lin, H.-T.: Ordinal regression by extended binary classification. Advances in Neural Information Processing Systems, 19:865, 2007.

Li, S. Z.: Markov random field modeling in image analysis. Springer Science & Business Media, 2009.

Lichman, M.: UCI machine learning repository, 2013.

Lin, H.-T.: From ordinal ranking to binary classification. Doctoral dissertation, California Institute of Technology, 2008.

Lin, H.-T.: 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.

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

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P.: Focal loss for dense object detection. IEEE transactions on pattern analysis and machine intelligence, 2018.

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

Liu, L., Sun, L., Rui, Y., Shi, Y., and Yang, S.: Web video topic discovery and tracking via bipartite graph reinforcement model. In Proceedings of the 17th International Conference on World Wide Web, pages 1009–1018. ACM, 2008.

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

Livni, R., Crammer, K., and Globerson, A.: A simple geometric interpretation of svm using stochastic adversaries. In Artificial Intelligence and Statistics, pages 722–730, 2012.

MacCartney, B., Galley, M., and Manning, C. D.: A phrase-based alignment model for natural language inference. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 802–811. Association for Computational Linguistics, 2008.

Manning, C. D. and Schütze, H.: Foundations of statistical natural language processing. MIT press, 1999.

McCullagh, P. and Nelder, J. A.: Generalized linear models, volume 37. CRC press, 1989.

Menegola, A., Fornaciali, M., Pires, R., Bittencourt, F. V., Avila, S., and Valle, E.: Knowledge transfer for melanoma screening with deep learning. In Biomedical Imaging (ISBI 2017), 2017 IEEE 14th International Symposium on, pages 297–300. IEEE, 2017.

Namkoong, H. and Duchi, J. C.: Stochastic gradient methods for distributionally robust optimization with f-divergences. In Advances in Neural Information Processing Systems, pages 2208–2216, 2016.

Namkoong, H. and Duchi, J. C.: Variance-based regularization with convex objectives. In Advances in Neural Information Processing Systems, pages 2971–2980, 2017.

Nesterov, Y.: Introductory Lectures on Convex Optimization: A Basic Course. Springer, 2003.

Nowozin, S., Lampert, C. H., et al.: Structured learning and prediction in computer vision. Foundations and Trends® in Computer Graphics and Vision, 6(3–4):185–365, 2011.

Osokin, A., Bach, F., and Lacoste-Julien, S.: On structured prediction theory with calibrated convex surrogate losses. In Advances in Neural Information Processing Systems, pages 301–312, 2017.

Padó, S. and Lapata, M.: Optimal constituent alignment with edge covers for semantic projection. In Proceedings of the International Conference on Computational Linguistics, pages 1161–1168. Association for Computational Linguistics, 2006.

Pearl, J.: Bayesian networks: A model of self-activated memory for evidential reasoning. In Proceedings of the 7th Conference of the Cognitive Science Society, 1985, 1985.

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

Petterson, J., Yu, J., McAuley, J. J., and Caetano, T. S.: Exponential family graph matching and ranking. In Advances in Neural Information Processing Systems, pages 1455–1463, 2009.

Purushotham, S., Meng, C., Che, Z., and Liu, Y.: Benchmarking deep learning models on large healthcare datasets. Journal of Biomedical Informatics, 83:112 – 134, 2018.

Rakhlin, A., Shvets, A., Iglovikov, V., and Kalinin, A. A.: Deep convolutional neural networks for breast cancer histology image analysis. In International Conference Image Analysis and Recognition, pages 737–744. Springer, 2018.

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

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

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

Regier, J., Miller, A., McAuliffe, J., Adams, R., Hoffman, M., Lang, D., Schlegel, D., and Prabhat, M.: Celeste: Variational inference for a generative model of astronomical images. In International Conference on Machine Learning, pages 2095–2103, 2015.

Rennie, J. D. M. and Srebro, N.: 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.

Rockafellar, R. T.: Convex Analysis. Princeton Mathematics Series. Princeton, NJ, Princeton University Press, 1970.

Rui, X., Li, M., Li, Z., Ma, W.-Y., and Yu, N.: Bipartite graph reinforcement model for web image annotation. In Proceedings of the 15th ACM International Conference on Multimedia, pages 585–594. ACM, 2007.

Sadeghian, A., Sundaram, L., Wang, D., Hamilton, W., Branting, K., and Pfeifer, C.: Semantic edge labeling over legal citation graphs. In Proceedings of the Workshop on Legal Text, Document, and Corpus Analytics (LTDCA-2016), pages 70–75, 2016.

Schmidt, M.: minConf: projection methods for optimization with simple constraints in Matlab. `http://www.cs.ubc.ca/~schmidtm/Software/minConf.html`, 2008.

Schmidt, M., Berg, E., Friedlander, M., and Murphy, K.: Optimizing costly functions with simple constraints: A limited-memory projected quasi-Newton algorithm. In Artificial Intelligence and Statistics, pages 456–463, 2009.

Shafieezadeh-Abadeh, S., Esfahani, P. M., and Kuhn, D.: Distributionally robust logistic regression. In Advances in Neural Information Processing Systems, pages 1576–1584, 2015.

Shalev-Shwartz, S., Singer, Y., Srebro, N., and Cotter, A.: Pegasos: Primal estimated subgradient solver for SVM. Mathematical Programming, 127(1):3–30, 2011.

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

Shi, Z., Zhang, X., and Yu, Y.: Bregman divergence for stochastic variance reduction: Saddle-point and adversarial prediction. In Advances in Neural Information Processing Systems, pages 6033–6043, 2017.

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

Sontag, D., Globerson, A., and Jaakkola, T.: Introduction to dual composition for inference. In Optimization for Machine Learning. MIT Press, 2011.

Steinwart, I. and Christmann, A.: Support Vector Machines. Springer Publishing Company, Incorporated, 1st edition, 2008.

Sutton, C., McCallum, A., et al.: An introduction to conditional random fields. Foundations and Trends® in Machine Learning, 4(4):267–373, 2012.

Tang, D., Qin, B., and Liu, T.: Aspect level sentiment classification with deep memory network. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 214–224, 2016.

Taskar, B., Chatalbashev, V., Koller, D., and Guestrin, C.: Learning structured prediction models: A large margin approach. In Proceedings of the International Conference on Machine Learning, pages 896–903. ACM, 2005.

Taskar, B., Lacoste-Julien, S., and Klein, D.: A discriminative matching approach to word alignment. In Conference on Human Language Technology and Empirical Methods in Natural Language Processing, pages 73–80. Association for Computational Linguistics, 2005.

Taylor, W. R.: Protein structure comparison using bipartite graph matching and its application to protein structure classification. Molecular & Cellular Proteomics, 1(4):334–339, 2002.

Teo, C. H., Vishwanthan, S. V. N., Smola, A. J., and Le, Q. V.: Bundle methods for regularized risk minimization. Journal of Machine Learning Research, 11:311–365, January 2010.

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

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

Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y.: Large margin methods for structured and interdependent output variables. Journal of Machine Learning Research, 6(Sep):1453–1484, 2005.

Tu, H.-H. and Lin, H.-T.: 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.

Valiant, L. G.: The complexity of computing the permanent. Theoretical Computer Science, 8(2):189–201, 1979.

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

Vapnik, V. N.: Statistical learning theory, volume 1. Wiley New York, 1998.

Vaswani, A., Bengio, S., Brevdo, E., Chollet, F., Gomez, A., Gouws, S., Jones, L., Kaiser, L., Kalchbrenner, N., Parmar, N., Sepassi, R., Shazeer, N., and Uszkoreit, J.: Tensor2tensor for neural machine translation. In Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers), pages 193–199. Association for Machine Translation in the Americas, 2018.

Vedaldi, A.: A MATLAB wrapper of SVM$^{\text{struct}}$. `http://www.vlfeat.org/~vedaldi/code/svm-struct-matlab`, 2011.

Villani, C.: Optimal transport: old and new, volume 338. Springer Science & Business Media, 2008.

Volkovs, M. and Zemel, R. S.: Efficient sampling for bipartite matching problems. In Advances in Neural Information Processing Systems, pages 1313–1321, 2012.

Von Neumann, J.: A certain zero-sum two-person game equivalent to the optimal assignment problem. Contributions to the Theory of Games, 2:5–12, 1953.

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

Wang, H., Xing, W., Asif, K., and Ziebart, B.: Adversarial prediction games for multivariate losses. In Advances in Neural Information Processing Systems, pages 2710–2718, 2015.

Wang, Y., Makedon, F., Ford, J., and Huang, H.: A bipartite graph matching framework for finding correspondences between structural elements in two proteins. In International Conference of the IEEE Engineering in Medicine and Biology Society, pages 2972–2975, 2004.

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

Xue, N. and Palmer, M.: Calibrating features for semantic role labeling. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, 2004.

Zafar, M. B., Valera, I., Gomez Rodriguez, M., and Gummadi, K. P.: Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In Proceedings of the 26th International Conference on World Wide Web, pages 1171–1180, 2017.

Zhang, T.: Statistical analysis of some multi-category large margin classification methods. Journal of Machine Learning Research, 5(Oct):1225–1251, 2004.

Zhu, J. and Hastie, T.: Kernel logistic regression and the import vector machine. In Advances in Neural Information Processing Systems, pages 1081–1088, 2002.

Zhu, J., Hoi, S. C., Lyu, M. R., and Yan, S.: Near-duplicate keyframe retrieval by nonrigid image matching. In Proceedings of the 16th ACM International Conference on Multimedia, pages 41–50. ACM, 2008.

VITA

NAME	Rizal Zaini Ahmad Fathony
EDUCATION	**PhD in Computer Science**, University of Illinois at Chicago, 2019
	MS in Computer Science, University of Illinois at Chicago, 2014
	BAS in Statistical Computing, Institute of Statistics, 2007
WORK EXPERIENCE	**Research Assistant**, University of Illinois at Chicago, Chicago, Illinois, 2014 - 2019
	Research Intern, Technicolor Research AI Lab, Los Altos, California, 2017
	Teaching Assistant, University of Illinois at Chicago, Chicago, Illinois, 2015 - 2016
	Statistical Dissemination System Developer, Central Bureau of Statistics Indonesia, Jakarta, Indonesia, 2007 - 2012
PUBLICATIONS	**Rizal Fathony**, Kaiser Asif, Anqi Liu, Mohammad Ali Bashiri, Wei Xing, Sima Behpour, Xinhua Zhang, Brian D. Ziebart, "Consistent robust adversarial prediction for general multiclass classification". *arXiv preprint arXiv:1812.07526*, 2018
	Rizal Fathony, Ashkan Rezaei, Mohammad Ali Bashiri, Xinhua Zhang, Brian D. Ziebart. "Distributionally Robust Graphical Models", *Advances in Neural Information Processing Systems (NeurIPS)*, 2018
	Rizal Fathony*, Sima Behpour*, Xinhua Zhang, Brian D. Ziebart, "Efficient and Consistent Adversarial Bipartite Matching", *International Conference on Machine Learning (ICML)*, 2018

Rizal Fathony, Mohammad Ali Bashiri, Brian D. Ziebart, "Adversarial Surrogate Losses for Ordinal Regression", *Advances in Neural Information Processing Systems (NIPS)*, 2017

Rizal Fathony, Anqi Liu, Kaiser Asif, Brian D. Ziebart, "Adversarial Multiclass Classification: A Risk Minimization Perspective", *Advances in Neural Information Processing Systems (NIPS)*, 2016

Anqi Liu, **Rizal Fathony**, Brian D. Ziebart, "Kernel Robust Bias-Aware Prediction under Covariate Shift", *ArXiv Preprints*, 2016