Adversarial Approach to

Cost-sensitive Classification and Sequence Tagging

BY

KAISER NEWAJ ASIF
B.S., Bangladesh University of Engineering and Technology, 2009

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
- Ajay Kshemkalyani
- Xinhua Zhang
- Sybil Derrible, Civil and Materials Engineering
- Didem Ozevin, Civil and Materials Engineering

To my parents

ACKNOWLEDGMENT

This thesis would not be possible without the support of many people.

First and foremost, I am thankful to my advisor Dr. Brian D. Ziebart, whose patience and guidance has enabled me to learn how to conduct and advance in research and eventually complete this thesis. I am also thankful to my committee members – Dr. Ajay Kshemkalyani, Dr. Xinhua Zhang, Dr. Sybil Derrible, and Dr. Didem Ozevin – for their valuable feedback to the thesis.

I would also like to thank my collaborators Sima Behpour, Wei Xing, and Jia Li for their contributions in discussion, research, help with manuscripts and experiments. I thank all my past and present lab-mates, especially Sanket Gaurav, Rizal Fathony, Sima Behpour, Xiangli Chen, Anqi Liu for their support, sharing knowledge, and overall having a good time together.

Thanks to my friends, including but not limited to, Adnan Mahmud, Rakib Hasan, Ragib Ahsan, Hasan Iqbal, and Abm Musa, for helping me through various difficult scenarios through out this journey.

Lastly, I thank my parents Abdul Mutalib and Monowara Begum for believing in me, and supporting my endeavor. Thanks to my brother Khaled Mahmud Arif and cousin Arifa Sharmin for their encouragements. And of course to my wife Nishat Jahan Tonni for her support and encouragements that helped me to go through the final steps of this journey.

ACKNOWLEDGMENT (Continued)

I should also acknowledge the grants that funded my research. Work on the cost-sensitive classification and the general framework development was supported by the National Science Foundation under Grant No. #1227495, *Purposeful Prediction: Co-robot Interaction via Understanding Intent and Goals*. The research in sequence tagging was supported in part by the NSF grants III-1514126 (Ziebart, Berger-Wolf), CNS-1248080 (Berger-Wolf), and RI-1526379 (Ziebart). The work in welding detection was supported by the Department of the Army through the Digital Manufacturing and Design Innovation Institute via the grant DMDII 15-14-03, *Intelligent Welding: Real Time Monitoring, Diagnosis, and Decision using Multi- sensor and Machine Learning*.

KNA

CONTRIBUTIONS OF AUTHORS

Dr. Brian D. Ziebart has actively helped with his suggestions, ideas, and reviews in all of the publications cited in this thesis.

Chapter 1 and chapter 2 contains parts of initial motivation for the approach and background research from the publication (Asif et al., 2015) where Dr. Ziebart helped with the discussion and writing. Wei Xing and Dr. Sima Behpour are co-authors in (Asif et al., 2015), where they have contributed to the background study of cost-sensitive learning. Dr. Jia Li helped in the background study of sequence tagging from (Li* et al., 2016).

Chapter 3 contains the theory from (Asif et al., 2015) and (Li* et al., 2016). I was the first author in (Asif et al., 2015). Wei Xing and Dr. Behpour participated in discussion and proof-reading the manuscript. Both Dr. Li and I had equal contribution in (Li* et al., 2016), I formulated the single oracle algorithm in the paper. Dr. Li formulated the double oracle and best response algorithms.

Chapter 4 contains results from the papers. Wei Xing and Dr. Behpour contributed with the experiments of other methods in (Asif et al., 2015). FAQ segmentation result is taken from (Li* et al., 2016), where I helped with the experiments.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

CHAPTER **PAGE**

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (Continued)

FIGURE **PAGE**

SUMMARY

In many machine learning applications, predicting incorrect classes or labels incurs different penalties depending on the predicted class and the actual class. Cost-sensitive classification formulates this situation by seeking predictions that minimize this variable loss. Since directly optimizing the empirical cost-sensitive loss is generally intractable, existing cost-sensitive methods minimize surrogate loss functions. For example, the support vector machine (SVM) uses the hinge loss. However, the SVM can fail to learn the cost-minimizing prediction for even in ideal learning conditions (i.e., it does not provide a Fisher consistency guarantee). On the other hand, Logistic Regression, which uses log-loss as the surrogate, is difficult to adapt to the cost-sensitive setting. Although it allows class importance weights to be incorporated, it cannot be adapted to the cost-sensitive setting with more than two classes.

We formulate the cost-sensitive classification as a minimax game between a predictor and a hypothetical adversary who approximates the training data labels, but is constrained by some training data properties. We directly include the cost-sensitive loss measure instead of a surrogate loss in the formulation. Unlike empirical risk minimization the resulting optimization problem is convex, allowing us to efficiently solve it. We develop and apply this method for multiclass cost-sensitive classification with arbitrary cost matrices. We then extend this work to sequence tagging (multiple interrelated variables) where Hamming loss is considered as the cost of mismatch between the target sequence and the predicted sequence. Later we improve the sequence tagging algorithm for faster computation and compare the two methods. We discuss

SUMMARY (Continued)

a real-world application to welding quality detection and activity recognition. We demonstrate that this adversarial approach is competitive with traditional methods, while having the theoretical benefit of consistency.

CHAPTER 1

INTRODUCTION

(Contents of this chapter were published in Asif, Kaiser, Wei Xing, Sima Behpour, and Brian D. Ziebart. "Adversarial Cost-Sensitive Classification." In *UAI*, pp. 92-101. 2015. (Asif et al., 2015))

In machine learning, classification is a task where different entities fall into different categories and based on the attributes of the entities, an algorithm learns to categorize future unseen entities. In a general classification task, the learning algorithm incurs a penalty whenever a predicted label does not match the corresponding actual label. For example, if a model is to classify cat versus dog pictures, it accumulates one misclassification cost whenever it identifies a dog picture as a cat or vice versa. However, in many applications of machine learning, the penalty or cost for classification errors is not uniform and instead depends on both the predicted label and the actual label. For example, an incorrect disease diagnosis may lead to treatments that cause complications of varying severity depending on the patient's actual disease. These different incurred penalties for mistakes can be represented as a confusion cost matrix that is indexed by the predicted class (row) and actual class (column). As shown in the following confusion cost matrix for a classification task with four possible labels,

$$\mathbf{C} = \begin{bmatrix} 0 & 1 & 2 & 0 \\ 3 & 0 & 1 & 3 \\ 4 & 2 & 0 & 1 \\ 1 & 1 & 2 & 0 \end{bmatrix}, \tag{1.1}$$

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

For an illustrative example, let us assume a binary classification problem of identifying expired items in a retail store based on some specific attribute – if the item is predicted to be expired then it will be removed from the shelf. Let us also assume that history shows that an approximate proportion of 0.3 of the items have the specific attribute are actually expired, rest are not. Then a classifier will predict future items having that attribute to be “not-expired” and be wrong at frequency 0.3 for the truly expired item, which is lower than the opposite choice. Now, if an item is falsely identified to be expired its value is the incurred cost by this prediction, but if an expired item is bought by a customer it results into a higher cost due to customer dissatisfaction which, let us suppose, has been quantified to be three times higher than the cost of discarding a good item. Then the confusion cost can be written as shown in Table I. If we

TABLE I: Costs for misclassification of item's expiration.

	not-expired	expired
predict not-expired	0	3
predict expired	1	0

use the classifier as mentioned above, which misclassifies the items with frequency 0.3, which are in fact "expired," to be "not-expired," the average cost will be $0.3 \times 3 = 0.9$ unit. On the other hand if this expected cost is taken into account, then classifying future items having that attribute to be expired will cause $0.7 \times 1 = 0.7$ unit average cost. Therefore, although general classification method suggests a "not-expired" prediction, predicting "expired" based on the cost-sensitive decision incurs smaller expected cost.

Other applications of cost-sensitive classification include imbalanced class distributions, where during the training process smaller class is given higher misclassification penalty so that overall loss of the full data-set accounts for all categories in a balanced way. In fraud detection, missing a fraud might have a higher cost than falsely labeling a valid instance as fraud. A similar situation applies to bank loan approval, where approving a bad customer has a higher risk than rejecting a good customer (Elkan, 2001). In time-series prediction one may vary the cost based on timeliness (Turney, 2002). There are three main ways cost-sensitive classification can be performed:

1. Learn the class probability distribution and use the Bayes optimal decision to select the class that minimizes expected cost under the estimated distribution;

2. Reweight or relabel training samples and train cost-insensitive classifiers based on the modified dataset; or
3. Train a classifier that directly incorporates the cost in its design.

A natural goal for machine learning is to obtain a classifier that minimizes the expected cost incurred when classifying an example. Previous research in cost-sensitive learning primarily takes existing classification methods based on **empirical risk minimization** and tries to adapt them in various ways to be sensitive to these misclassification costs. Reweighting methods artificially augment the training data with copies of "high cost" examples to make the classifier more cost-sensitive to them (Chan and Stolfo, 1998; Elkan, 2001; Zadrozny et al., 2003; Zhou and Liu, 2010). While this is trivial for the binary classification problem, for the multiclass problem, the costs need to have some specific consistency that allows similar reweighting ratio across all pair of classes (Zhou and Liu, 2010). Other methods modify the criteria used to obtain a classifier that incorporates mistake-specific losses (Knoll et al., 1994; Turney, 1995; Elkan, 2001; Brefeld et al., 2003; Ling et al., 2004; Lomax and Vadera, 2013). However, in both cases the non-convexity of the cost-sensitive loss function makes direct empirical risk minimization impractical (Hoffgen et al., 1995).

To avoid these difficulties, surrogate loss functions that are convex (e.g., the hinge loss) are instead minimized. These include the hinge loss and the log loss surrogate loss functions for classification tasks. However, these approximations using surrogate losses can introduce significant suboptimality. On the other, hand modifying the surrogate loss is a difficult task when given arbitrary cost matrices, because the loss incurred is typically non-convex in the classifier's parameters. For example, support vector machines (Cortes and Vapnik, 1995), have been successfully demonstrated for a number of classification tasks, but efforts to make them cost-sensitive have been restricted to cost matrices with specific consistency properties as mentioned above (Zhou and Liu, 2010).

In this thesis, we develop and explore the benefits of a different approach: **adversarial prediction methods for cost-sensitive learning.** Rather than integrating cost-sensitivity into existing machine learning techniques, we develop an approach from first principles to robustly minimize the expected cost. Our approach treats classifier construction as a game against an adversarial evaluator (Topsøe, 1979; Grünwald and Dawid, 2004). This enables us to directly minimize the cost-sensitive loss on an approximation of the training data instead of using a convex approximation of the cost-sensitive loss, as is done with empirical risk minimization. Inference reduces to solving a zero-sum game in our approach. This is efficiently accomplished using linear programming. We obtain parameter estimates by constructing game payoff parameters using convex optimization methods. The key benefit of our approach is that the exact confusion cost matrix is employed rather than a convex surrogate.

Similar cost-sensitive losses arise in more complicated structured prediction tasks as well. In a sequence tagging task, a sequence of variables are labeled jointly that may depend on the labels of each other. Sequence tagging is used in Natural Language Processing (NLP) for Parts of Speech (POS) tagging (Lafferty et al., 2001b; Sha and Pereira, 2003) where words are sequence of variables and target class for each word is the POS. Then, if a word can be labeled as more than one POS, its label also depends on adjacent labels. For example, a preposition is always followed by a noun clause and the word “on” can be either preposition, adjective or adverb. Various activity recognition task also falls under sequence tagging. In human activity recognition, human activities are tracked using smartphone sensor data (Reyes-Ortiz et al., 2016) as they transition from sitting to standing to walking and so on. In some applications, cost-sensitive sequence tagging may be preferred. For example, in POS tagging, labeling “physics” as plural noun NNS instead of singular NN is a minor error compared to labeling “plans” as NNS instead of verb VBZ (Song et al., 2012). In case of activity recognition, walking vs walking-downstairs might have a higher penalty than sit-to-stand vs lie-to-stand. Existing methods for sequence tagging also optimize similar surrogate loss functions. For example, log-loss is optimized in the conditional random field (CRF) and the hinge loss is minimized in the structured support vector machine (structured-SVM). To perform a cost-sensitive sequence tagging, structured-SVM can be modified (Song et al., 2012). However, we extend our adversarial framework to sequence tagging which does not require us to assume a surrogate loss function and also can easily incorporate cost-sensitivity.

In Chapter 2, we discuss related work and provide a brief introduction to background material, including empirical risk minimization, cost-sensitive classification, adversarial approaches, and game theory. We develop the adversarial approach for classification in Chapter 3 and illustrate its procedures with toy examples. We extend it to sequence tagging, and develop and compare two approaches to solve the optimization problem for the sequence tagging task. We discuss results for both cost-sensitive classification and sequence tagging in Chapter 4. We elaborate using an application of sequence tagging to automated welding quality detection. Finally, in Chapter 5 we provide our conclusion, and discuss future directions.

CHAPTER 2

RELATED WORK

(Sections of this chapter were published in Asif, Kaiser, Wei Xing, Sima Behpour, and Brian D. Ziebart. "Adversarial Cost-Sensitive Classification." In *UAI*, pp. 92-101. 2015. (Asif et al., 2015), and in Jia Li, Kaiser Asif, Hong Wang, Brian D. Ziebart, and Tanya Y. Berger-Wolf. "Adversarial Sequence Tagging." In *IJCAI*, pp. 1690-1696. 2016. (Li* et al., 2016))

2.1 Empirical Risk Minimization

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

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

with a regularization penalty $\lambda||\theta||$ often added to avoid overfitting to available training data. Here we denote scalar values and vector values lowercase non-bold, x, and bold, $\mathbf{x}$, and random variables in capital X or $\mathbf{X}$. Unfortunately, many combinations of classification functions, $f_\theta(\mathbf{x})$, and loss functions, $\text{loss}(\cdot, \cdot)$, do not lend themselves to efficient parameter optimization under

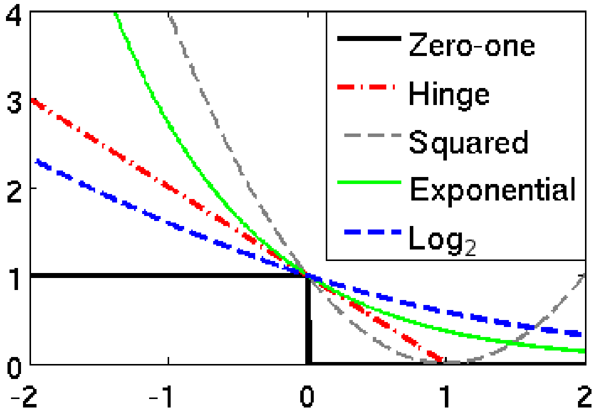


Figure 1: Convex surrogates for the zero-one loss.

the empirical risk minimization (ERM) formulation. For example, minimizing the zero-one loss measuring the misclassification rate,

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

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 1) that serve as upper bounds on the desired loss function are often used to create tractable optimization problems. The popular support vector machine (SVM) classifier (Cortes and Vapnik, 1995), for example, employs the hinge-loss—an upper bound on the zero-one loss—to avoid the often intractable empirical risk minimization problem. For binary classification with a scoring function $\psi_{\theta,\mathbf{x}} =$

$\theta^{\mathrm{T}}\phi(\mathbf{x})$ where sign of the score defines the class, and target class $y \in \{-1, +1\}$ the hinge loss is defined as

$$\mathrm{loss}(y) = \max(0, 1 - y\psi_{\theta,\mathbf{x}}).$$

A generalized version for more than two class is given by $\mathrm{loss}(y) = \max(0, 1 + \max_{y \neq y'} \theta^{\mathrm{T}}\phi(\mathbf{x}, y') - \theta^{\mathrm{T}}\phi(\mathbf{x}, y))$ (Crammer and Singer, 2002). SVM for binary classification optimizes:

$$\min_{\theta,b,\xi_t} \frac{1}{2}\theta^{\mathsf{T}}\theta + C\sum_{t=1}^{m} \xi_t \tag{2.2}$$

$$\text{subject to: } \xi_t \geq 0, t = 1, \dots, m$$

$$y_t\left(\theta^{\mathsf{T}}\phi(x_t) + b\right) \geq 1 - \xi_t^i$$

which is essentially $\min\left(\frac{1}{2}||\theta|| + C\sum_{t=1}^{m} \max\left[0, 1 - y_t\left(\theta^{\mathsf{T}}\phi(x_t) + b\right)\right]\right)$.

Another popular classifier is Adaboost (Freund and Schapire, 1997), it incrementally minimizes the exponential loss,

$$\mathrm{loss}(y) = e^{-yf_m(x)},$$

where $f_m(x)$ is the incrementally updated predictor.

The difference between these convex surrogates and the actual loss can introduce a substantial mismatch between optimal parameter estimation under the surrogate loss function and optimal parameter estimates for the original performance objective.

2.2 Cost-sensitive Learning

Cost-sensitive learning considers more general loss functions than the zero-one loss in which the loss depends on the actual and the predicted class. One approach is to estimate the conditional label distribution, $\hat{\mathsf{P}}(\mathsf{y}|\mathbf{x})$, and employ the Bayesian optimal classifier:

$$\hat{\mathsf{f}}(\mathbf{x}) = \underset{\mathsf{y}' \in \mathcal{Y}}{\operatorname{argmin}}\, \mathbb{E}_{\hat{\mathsf{P}}(\mathsf{y}|\mathbf{x})}[\mathsf{C}_{\mathsf{y}',\mathsf{Y}}], \tag{2.3}$$

using, e.g., the cost matrix of Equation 1.1. However, accurately estimating the conditional label distribution will typically require much more data than methods that directly learn the best class prediction for a given loss function (Margineantu, 2002).

Early meta-learning methods for cost-sensitive learning attempt to modify how a cost-insensitive learner is used during training and/or prediction time so that the end result of its use is cost-sensitive. One approach for this is to either stratify or reweight available training data so that more costly mistakes will incur a larger overall cost and therefore the resulting classifier will be more sensitive to them (Chan and Stolfo, 1998; Elkan, 2001; Zadrozny et al., 2003; Zhou and Liu, 2010). For binary classification with confusion cost of:

	actual negative	actual positive
predict negative	c_{00}	c_{01}
predict positive	c_{10}	c_{11}

Equation 2.3 implies the cost-sensitive prediction is class $+1$ if and only if

$$(1-p)c_{10} + pc_{11} \leq (1-p)c_{00} + pc_{01}$$

where $p = P(y = +1|\mathbf{x})$ is the cost-insensitive probability of sample x being of class $+1$. Rearranging this equation gives the threshold p^* (instead of the 0.5, i.e. assign class $+1$ if $p \geq p^*$) as

$$p^* = \frac{c_{10} - c_{00}}{c_{10} - c_{00} + c_{01} - c_{11}} \tag{2.4}$$

The number of negative samples are then multiplied by $p^*/(1-p^*)$ (Elkan, 2001). With $c_{00} = c_{11} = 0$, i.e. correct classification having 0 cost, multiplication factor for number of negative samples is:

$$\frac{p^*}{1-p^*} = \frac{c_{10}}{c_{01}}$$

However, the validity of this approach is limited to a restricted class of *consistent* cost matrices when applied to multi-class prediction tasks (Domingos, 1999; Zhou and Liu, 2010). The simpler traditional multiclass reweighting is

$$\frac{w_i}{w_j} = \frac{\mathsf{cost}_i}{\mathsf{cost}_j} = \frac{\sum_k c_{ik}}{\sum_k c_{jk}} \text{ ,}$$

which certainly is not ideal since it does not retains the misclassification cost ratio, i.e. c_{ij}/c_{ji} (Zhou and Liu, 2010). Therefore Zhou and Liu proposed to use rescaling only if m weights $\mathbf{w} = [w_1, w_2, ..., w_m]^T$ can be computed from the cost matrix such that

$$\frac{w_i}{w_j} = \frac{\mathrm{cost}_{ik}, \forall k}{\mathrm{cost}_{jl}, \forall l},$$

otherwise learn $m(m-1)/2$ binary cost-sensitive classifiers and use voting to get a final prediction. A method that reduces multi-class predictions to binary predictions using iterative reweighting, data space expansion, and gradient boosting with stochastic ensembles (Abe et al., 2004) has been proposed to overcome these limitations. The *Metacost* algorithm (Domingos, 1999) similarly wraps around any underlying classifier. It uses bagging (Breiman, 1996) to produce label probability estimates, which it then uses to modify training data labels using Equation 2.3, and retrain the classifier with the modified training labels to produce cost-sensitive learner. The *cost-transformation technique* (Lin, 2008) modifies the objective during the training and uses traditional classifiers to perform cost-sensitive classification.

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

Boosting iteratively creates an ensemble of weak classifiers that are then combined to create a much stronger classifier (Freund and Schapire, 1997) that often performs well in practice. Cost-sensitive boosting techniques employ cost-sensitive weak learners to produce a stronger learner that is cost-sensitive as well (Fan et al., 1999; Ting, 2000). This is accomplished by minimizing the risk over the training dataset, $\frac{1}{n}\sum_{i=1}^{n}\text{loss}'(C, y_i, S(\mathbf{x}_i))$, using a generalized surrogate loss function, $\text{loss}'(C, \tilde{y}, S_m(\mathbf{x}))$, for the cost matrix C, class label $\tilde{y}$, and where $S_y(\mathbf{x})$ represents the classifier confidence in assigning class y to data point $\mathbf{x}$. Recently developed loss functions are the Generalized Exponential Loss (GEL), $\sum_{y'} C_{y,y'} e^{S_{y'}(\mathbf{x})-S_y(\mathbf{x})}$ and the Generalized Logistic Loss (GLL), $\log(1 + \sum_{y'} C_{y,y'} e^{S_{y'}(\mathbf{x})-S_y(\mathbf{x})})$. These loss functions are guess-averse and produce state-of-the-art performance when used in boosting for cost-sensitive classification (Beijbom et al., 2014).

Support vector machines (Cortes and Vapnik, 1995) have been generalized in the binary classification setting by penalizing mistakes for one class more than for the other class (Brefeld et al., 2003). Multiclass problems are reduced to binary classifiers using one-versus-all (Bottou et al., 1994) and one-versus-one (Knerr et al., 1990) prediction tasks. The *Cost-Sensitive One-*

Versus-All (CSOVA) algorithm (Lin, 2008) trains a separate binary SVM classifier for each class:

$$\min_{\theta^i, b^i, \xi_t^i} \frac{1}{2}(\theta^i)^T\theta^i + C\sum_{t=1}^{t} w_t \xi_t^i \tag{2.5}$$

$$\begin{aligned} \text{subject to: } & \xi_t^i \geq 0, t = 1, \ldots, m \\ & (\theta^i)^T\phi(x_t) + b^i \geq 1 - \xi_t^i, \text{ if } y_t = i \\ & (\theta^i)^T\phi(x_t) + b^i \leq -1 + \xi_t^i, \text{ if } y_t \neq i, \end{aligned}$$

where w_t is a weight based on the misclassification cost for example t. Prediction is done based on maximum prediction score of the predictors. The *Cost-Sensitive One-Versus-One* (CSOVO) algorithm (Lin, 2010) instead constructs a total of $m(m-1)/2$ classifiers—one for each pair of classes (i, j):

$$\min_{\theta^{i,j}, b^{i,j}, \xi_t^{i,j}} \frac{1}{2}(\theta^{i,j})^T\theta^{i,j} + C\sum_{t} w_t \xi_t^{i,j} \tag{2.6}$$

$$\begin{aligned} \text{subject to: } & \xi_t^{i,j} \geq 0, t = 1, \ldots, m \\ & (\theta^{i,j})^T\phi(x_t) + b^{i,j} \geq 1 - \xi_t^{i,j} \text{ if } y_t = i \\ & (\theta^{i,j})^T\phi(x_t) + b^{i,j} \leq -1 + \xi_t^{i,j} \text{ if } y_t = j. \end{aligned}$$

Prediction is done based on voting. Using structured SVM methods (Tsochantaridis et al., 2005) to directly incorporate cost-sensitivity into the multiclass generalization of the hinge loss (Lee et al., 2004),

$$\min_{\theta,\, \epsilon \geq \mathbf{0}} \theta \cdot \theta + \alpha \sum_i \epsilon_i \text{ such that:} \tag{2.7}$$

$$\theta \cdot \phi(\mathbf{x}_i, y_i) - \theta \cdot \phi(\mathbf{x}_i, y') \geq C_{y', y_i} - \epsilon_i, \forall i, y' \neq y_i,$$

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

2.3 Consistency

A classifier is called Bayes optimal if it minimizes the probability error of the true distribution,

$$Y|x = \operatorname*{argmin}_y (1 - P(y|x)) = \operatorname*{argmax}_y P(y|x).$$

For unequal misclassification costs, this becomes,

$$Y|x = \operatorname*{argmin}_y \sum_{y'} C(y, y') P(y'|x).$$

Fisher consistency validates this theoretical property for a classifier.

Definition 1. *Given an arbitrarily rich feature representation* $\phi(\mathbf{x}, y)$*, a predictor* $\hat{f}(\mathbf{X})$ *trained on the true evaluation distribution* $P(\mathbf{X}, Y)$ *that minimizes the expected loss* $\mathbb{E}_{P(\mathbf{X},Y)}[\Delta(\hat{f}(\mathbf{X}), Y)]$ *is called* ***Fisher Consistent*** *for the loss function* $\Delta(\hat{f}(\mathbf{X}), Y)$*.*

We show that multiclass SVM is not Fisher consistent. Assume a cost-matrix $(C_{y_{prediced}, y_{true}})$ and true conditional distribution for the sample $\mathbf{x}$,

$$\mathbf{C} = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 1 & 2 & 0 \end{bmatrix}, \qquad P(Y_{true}|\mathbf{x}) = \begin{bmatrix} 0.4 \\ 0.25 \\ 0.35 \end{bmatrix},$$

then a Bayes optimal classifier will predict,

$$\operatorname*{argmin}_{y_{predicted}} \mathbf{C}P(Y_{true}|\mathbf{x}) = \operatorname*{argmin}_{y_{predicted}} \begin{bmatrix} 0.95 & 0.75 & 0.9 \end{bmatrix}^T = 2$$

In SVM, the optimal minimizer ψ^* minimizes the expected hinge loss

$$\mathbb{E}_{P(Y|\mathbf{x})} \left[\max_{y \neq y'} \left[C_{y,y'} + \psi(\mathbf{x}, y') - \psi(\mathbf{x}, y) \right]_+ \right]$$

To be Bayes optimal, the potentials should be at least $\psi(\mathbf{x}, Y) = [0, \delta, 0], \delta > 0$. Then expected loss for the given cost-matrix becomes,

$$
\begin{aligned}
&0.4 \times (\max(1+\delta, 2) - 0) + 0.25 \times (1 + 0 - \delta)_+ + 0.35 \times (2 + \delta - 0) \\
&= \begin{cases} 0.8 + 0.25 - 0.25\delta + 0.7 + 0.35\delta & 0 < \delta \leq 1 \\ 0.4 + 0.4\delta + 0 + 0.7 + 0.35\delta & \delta \geq 1 \end{cases} \\
&= \begin{cases} 1.75 + 0.1\delta & 0 < \delta \leq 1 \\ 1.1 + 0.75\delta & \delta \geq 1 \end{cases}
\end{aligned}
$$

This expected loss is larger than a degenerate case where all the potentials are equal, $\psi(\mathbf{x}, Y) = [0, 0, 0]$, and loss $= 0.4 \times (2 + 0 - 0) + 0.25 \times (1 - 0 + 0) + 0.35 \times (2 + 0 - 0) = 1.75$. Therefore, SVM for multiclass cost-sensitive classification is not Fisher consistent. But as we will see, by definition, our adversarial method is Fisher consistent.

2.4 Adversarial Methods

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

Our approach differs substantially from adversarial machine learning formulations that are made robust to adversarial shifts in the dataset (Dalvi et al., 2004; Liu and Ziebart, 2014) or uncertainty in the loss function (Wang and Tang, 2012) where minimax formulation is used to learn optimal classifier from a set of possible cost-matrices using existing cost-sensitive classification methods. We assume training and testing data are independent and identically distributed (IID) and the cost-sensitive loss function is fully known. We restrict our uncertainty to the conditional label distribution $\mathsf{P}(\mathsf{y}|\mathbf{x})$ and adversarially estimate it. In contrast with minimax approaches to classification that assume parametric forms of the data (Lanckriet et al., 2003), our approach allows the estimation of any conditional label distribution. Only training data properties are incorporated in the form of constraints on the adversary's conditional label distribution (Grünwald and Dawid, 2004). Our method is different from generative adversarial networks (GAN) (Goodfellow et al., 2014) which also use the minimax game but the goal there is to train a generative model as the max-player of the game that tries to mimic true training samples to fool the discriminator network which is the min-player trying to distinguish true samples from generated samples. For special cases of cost-sensitive cost matrices, our adversarial approach ca be analyzed, leading to more comparable algorithms. For example, 0-1 loss is the special case where costs along diagonal of the cost-matrix is 0 and non-diagonals are 1. In ordinal regression problem, the cost increases gradually as the distance from diagonal increases. Our adversarial approach can be extended to such cost-matrices easily and can achieve competitive performance (Fathony et al., 2016; Fathony et al., 2017).

2.5 Game Theory

Adversarial approach are motivated by game theory. In game theory, a game involves multiple players or decision makers where each individual's goal is to maximize a benefit or minimize a loss that is dependent upon their interactions. Without the knowledge of other players' actions, the decisions are made under uncertain conditions (Ferguson, 2014). We only utilize two-player zero-sum game theory. In a **two-player zero-sum** game, there are exactly two players, one player (Player I) gains as the other player (Player II) loses the exact amount, and the sum of the payoffs is zero, hence is the name "zero-sum". Mathematically a two player zero-sum game can be expressed by a strategic form.

Definition 2. *The **strategic form**, or normal form, of a two-person zero-sum game is given by a triplet* $(\mathsf{X}, \mathsf{Y}, \mathsf{A})$*, where*

1. X *is the nonempty set of strategies of Player I*
2. Y *is the nonempty set of strategies of Player II*
3. $\mathsf{A}\colon \mathsf{X} \times \mathsf{Y} \to \mathbb{R}$ *gives the payoff for each strategy pair.*

Each action or choice by a player separately is called **pure strategy** and when a combination of corresponding strategies with some randomness are selected by a player, it is called **mixed**

strategy. The optimal choice of each player's strategies is given the by the minimax theorem of von Neumann (von Neumann and Morgenstern, 1947) when both X and Y are finite.

Theorem 1. *The* ***Minimax Theorem*** *states that, for a finite two-person zero-sum game, there is a game value that is the minimum value Player I can ensure to win by a mixed strategy irrespective of Player II's actions. The game value is also the upper bound that Player II can ensure to lose via its mixed strategy.*

Let, $\mathbf{A}$ be a payoff matrix where $a_{i,j}$ is the payoff from Player II to Player I when Player I chooses row i and Player II chooses column j. $\mathbf{p} = (p_1, p_2, ..., p_m)$ be a mixed strategy where Player I chooses row i with probability p_i, and similarly Player II has a mixed strategy $\mathbf{q} = (q_1, q_2, ..., q_n)$. The value of the game is $V = \sum_i \sum_j p_i a_{ij} q_j = \mathbf{p^T A q}$. For $\mathbf{p}$ to be optimal, the following must be true,

$$\sum_i p_i a_{ij} \geq V, \qquad \forall j \tag{2.8}$$

that is, whichever column Player II chooses, Player I can ensure at least the expected game value V. Player II tries to minimize its loss. And therefore it will have V as the upper bound,

$$\sum_j a_{ij} q_j \leq V, \qquad \forall i. \tag{2.9}$$

When both players choose optimally, we have

$$\begin{aligned}
V &= \sum_j V q_j && \sum_j q_j = 1, \text{ probability distribution} \\
&\le \sum_j \left(\sum_i p_i a_{ij} \right) q_j && \text{Equation (Equation 2.8)} \\
&= \sum_i \sum_j p_i a_{ij} q_j = \sum_i p_i \left(\sum_j a_{ij} q_j \right) && \\
&\le \sum_i p_i V && \text{Equation (Equation 2.9)} \\
&= V && \text{(2.10)}
\end{aligned}$$

So, the optimal game value for both players are equal and unique.

We can follow Linear Programming to prove this theorem. If Player I moves first, its objective is,

$$\max_{\mathbf{p}} \min_{\mathbf{q}} \mathbf{p}^{\mathbf{T}} \mathbf{A} \mathbf{q} = \max_{\mathbf{p}} \min_{\mathbf{q}} \sum_{i=1}^{m} \sum_{j=1}^{n} p_i a_{ij} q_j. \tag{2.11}$$

Similarly Player II's objective is,

$$\min_{\mathbf{q}} \max_{\mathbf{p}} \mathbf{p}^{\mathbf{T}} \mathbf{A} \mathbf{q} = \min_{\mathbf{q}} \max_{\mathbf{p}} \sum_{i=1}^{m} \sum_{j=1}^{n} p_i a_{ij} q_j. \tag{2.12}$$

We have to show Equation 2.11 and Equation 2.12 are equal, then the value of the objective will be the game value and this will prove the minimax theorem.

Now let us observe that if $\mathbf{q}$ is known, then

$$\max_{\mathbf{p}} \sum_{i=1}^{m} \sum_{j=1}^{n} p_i a_{ij} q_j = \max_{1 \le i \le m} \sum_{j=1}^{n} a_{ij} q_j. \tag{2.13}$$

This is true because the right side is a pure strategy, the probabilistic mean on the left side is less or equal to the right, conversely, right is less or equal to the left since a pure strategy is a special case of mixed strategy and if left is not greater or equal, we can always find a distribution that chooses the maximum row from the right. Similarly if $\mathbf{p}$ is known,

$$\min_{\mathbf{q}} \sum_{i=1}^{m} \sum_{j=1}^{n} p_i a_{ij} q_j = \min_{1 \le j \le n} \sum_{i=1}^{m} p_i a_{ij}. \tag{2.14}$$

We can rewrite Equation 2.11 as,

$$\max_{\mathbf{p}} \min_{\mathbf{q}} \mathbf{p}^{\mathbf{T}} \mathbf{A} \mathbf{q} = \max_{\mathbf{p}} \min_{\mathbf{q}} \sum_{i=1}^{m} \sum_{j=1}^{n} p_i a_{ij} q_j = \max_{\mathbf{p}} \min_{1 \le j \le n} \sum_{i=1}^{m} p_i a_{ij}, \tag{2.15}$$

where $\sum_{i=1}^{m} p_i = 1$ and $p_i \ge 0$, since $\mathbf{p}$ is a probability distribution. To make the objective (right side of Equation 2.15) linear, we introduce a new variable v,

$$\begin{aligned} \max_{\mathbf{p}, v} \quad & v \\ \text{s.t.} \quad & v \le \min_{1 \le j \le n} \sum_{i=1}^{m} p_i a_{ij}, \\ & \sum_{i=1}^{m} p_i = 1, \quad p_i \ge 0. \end{aligned} \tag{2.16}$$

Which is equivalent to,

$$\begin{aligned} \max_{\mathbf{p},v} \quad & v \\ \text{s.t.} \quad & v \leq \sum_{i=1}^{m} p_i a_{ij}, \quad \forall j \in \{1, ..., n\} \\ & \sum_{i=1}^{m} p_i = 1, \quad p_i \geq 0. \end{aligned} \tag{2.17}$$

Similarly, Player II's objective can be written as $\min_{\mathbf{q}} \max_{1 \leq i \leq m} \sum_{j=1}^{n} a_{ij} q_j$, and the corresponding linear program,

$$\begin{aligned} \min_{\mathbf{q},w} \quad & w \\ \text{s.t.} \quad & w \geq \sum_{j=1}^{m} a_{ij} q_j, \quad \forall i \in \{1, ..., m\} \\ & \sum_{j=1}^{n} q_i = 1, \quad q_i \geq 0. \end{aligned} \tag{2.18}$$

It is easy to show that Equation 2.17 and Equation 2.18 are dual of each other, where minimize and maximize switches, for each constraint in primal a variable is created in the dual, variables corresponding to equality constraint is unbounded (v and w here), inequality constraints becomes non-negativity constraints and vice versa. Hence the above objectives are equal, and therefore Equation 2.11 and Equation 2.12 are equal. This proves that the optimal game value of the players are equal,

$$\max_{\mathbf{p}} \min_{\mathbf{q}} \mathbf{p^T A q} = \min_{\mathbf{q}} \max_{\mathbf{p}} \mathbf{p^T A q}.$$

We also use these linear programming formulations for solving the objectives in our algorithm later.

2.6 Lagrange Duality

We briefly discuss the Lagrangian method for convex optimization. For further details, readers are referred to the book Convex Optimization (Boyd and Vandenberghe, 2004). A set is called **convex** if for any line connecting any two points of the set completely lies within the set, i.e. for $x_1, x_2 \in C$ and $0 \leq \gamma \leq 1$, $\gamma x_1 + (1-\gamma)x_2 \in C$.

Definition 3. *A function* $f : \mathbf{R}^n \rightarrow \mathbf{R}$ *is* ***convex*** *if the domain of* f *is a convex set* C *and if for all* $x, y \in C$*, and* γ *with* $0 \leq \gamma \leq 1$*, we have*

$$f(\gamma x + (1-\gamma)y) \leq \gamma f(x) + (1-\gamma)f(y)$$

A function f is **concave** if $-f$ is convex.

The **Lagrangian** of an optimization is the constraints-augmented objective. Let an optimization be,

$$\begin{aligned} &\text{minimize} && f_0(x) \\ &\text{subject to} && f_i(x) \leq 0, \quad i = 1, ..., k \\ &&& h_i(x) = 0, \quad i = 1, ..., l \end{aligned}$$

Then the Lagrangian will be,

$$L(x, \eta, \theta) = f_0(x) + \sum_{i=1}^{k} \eta_i f_i(x) + \sum_{i=1}^{l} \theta_i h_i(x),$$

and the vectors η and θ are called Lagrange multiplier vectors. The function $g(\eta, \theta) = \inf_x L(x, \eta, \theta)$ is called the **Lagrange dual function**. This is a concave function as it is pointwise infimum of an affine function of (η, θ). This dual function provides the lower bounds of the original optimization. The maximum value of the dual, achieved by the dual problem,

$$\begin{aligned} \text{maximize} \quad & g(\eta, \theta) \\ \text{subject to} \quad & \eta \geq 0, \end{aligned}$$

gives the lower bound of the original optimization (minimize $f_0(x)$, called primal). The value of the dual and primal values are equal if strong duality holds. Then the optimizations are equivalent,

$$\underset{\eta,\theta}{\text{maximize}}\,\underset{x}{\text{minimize}}\, L(x, \eta, \theta) = \underset{x}{\text{minimize}}\,\underset{\eta,\theta}{\text{maximize}}\, L(x, \eta, \theta)$$

There are several ways to establish strong duality. By weaker **Slater's condition** if $f_0, ..., f_k$ are convex, and the inequality constraints are strict if they are not affine, then we have an x in the relative interior of the solution space, and strong duality holds. And, according to **Sion's minimax theorem** (Sion, 1958), if the domain of x is a closed and convex subset of $\mathbb{R}^n$ and the domain of the dual variables (η, θ) is a convex subset of $\mathbb{R}^m$, the objective is (quasi-)convex on x (i.e. f_0) and concave on (η, θ), and they are (semi-)continuous, then strong duality holds as well. We will see that our optimization satisfy all of the Sion's conditions easily.

2.7 Sequence Tagging

In this thesis we explore an adversarial approach to cost-sensitive classification and sequence tagging. For sequence tagging without cost-sensitivity, the exact loss can be measured as the Hamming loss, which is the count of mismatched tags of the sequence. However, directly minimizing Hamming loss is NP-hard (Höffgen and Simon, 1992). Conditional random fields (CRFs) and structured support vector machines (SSVMs) are two prominent methods which minimize the empirical risk of a surrogate loss function in sequence tagging:

$$\operatorname*{argmin}_{\theta} \mathbb{E}_{\tilde{P}(\mathbf{x},\mathbf{y})\hat{P}_\theta(\hat{\mathbf{y}}|\mathbf{x})} \left[\text{loss}\left(\mathbf{Y}, \hat{P}_\theta(\cdot|\mathbf{x})\right)\right] + \lambda||\theta|| \tag{2.19}$$

$$\text{or} \operatorname*{argmin}_{\theta} \mathbb{E}_{\tilde{P}(\mathbf{x},\mathbf{y})} \left[\text{loss}\left(\mathbf{Y}, \hat{f}_\theta(\mathbf{X})\right)\right] + \lambda||\theta||. \tag{2.20}$$

In CRFs, the loss is a logarithmic loss, $-\log \hat{P}(\mathbf{y}|\mathbf{x})$, and an exponential random field model, e.g., $\hat{P}(\mathbf{y}|\mathbf{x}) \propto \exp(\theta \cdot \Phi(\mathbf{x}, \mathbf{y}))$ is also employed in Equation 2.19 (Lafferty et al., 2001a). For SSVMs (Tsochantaridis et al., 2004), the structured hinge loss is a convex approximation to the Hamming loss, $\Delta(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{t=1}^{T} I(\hat{y}_t \neq \tilde{y}_t)$,

$$\left[\max_{\mathbf{y}' \neq \mathbf{y}} \Delta(\mathbf{y}, \mathbf{y}') + \theta \cdot (\Phi(\mathbf{x}, \mathbf{y}') - \Phi(\mathbf{x}, \mathbf{y}))\right]_+, \tag{2.21}$$

where $[f(x)]_+ \triangleq \max(0, f(x))$, and a linear discriminant function, $\hat{f}_\theta(\mathbf{x}) = \operatorname{argmax}_{\hat{\mathbf{y}} \in \boldsymbol{\mathcal{Y}}} \theta \cdot \Phi(\mathbf{x}, \hat{\mathbf{y}})$, are employed in Equation 2.20. The loss function of each model is a convex upper bound on the Hamming loss, $\sum_{t=1}^{T} I(\hat{y}_t \neq \tilde{y}_t)$.

In a cost-sensitive tagging task using CRF, following Bayes optimal prediction formula can be used during prediction after the conditional probability distribution is learn via conventional method,

$$\underset{\hat{y}_{1:T}}{\operatorname{argmin}} \mathbb{E}_{\hat{P}(\hat{y}_{1:T}|\mathbf{x}_{1:T})} \left[\sum_{t=1}^{T} C_{\hat{y}_t, y_t} \right]. \tag{2.22}$$

In structured SVM, cost-matrix can directly define the loss additively, $\Delta(\mathbf{y}, \mathbf{y}_i) = \sum_{t=1}^{T} C_{y_t, y_{i,t}}$:

$$\min_{\theta, \xi} \|\theta\|^2 + \alpha \sum_i \xi_i \tag{2.23}$$

such that,

$$(\theta \cdot \phi(\mathbf{x}, \mathbf{y}_i)) - (\theta \cdot \phi(\mathbf{x}, \mathbf{y})) \geq \Delta(\mathbf{y}, \mathbf{y}_i) - \xi_i$$

$$\xi_i \geq 0, \forall i, \forall \mathbf{y} \in \mathcal{Y}.$$

But, as mentioned previously, surrogate losses may be loose and hence may not always provide optimal parameters. Moreover, CRF usually has a higher computational cost. Therefore, we apply our adversarial formulation in sequence tagging as well.

2.8 Viterbi Algorithm

The Viterbi algorithm is a dynamic programming algorithm that finds the most probable sequence under the assumption of Markov independence (Viterbi, 1967). The Markov independence property states that the current state is independent of all other previous states if the immediate previous state is known. Viterbi algorithm finds the most probable labels $y_1, y_2, ..., y_T$ for observation $x_1, x_2, ..., x_T$ given emission probability distributions $P(x_t|y_t)$ (i.e. probability of observing x_i in state y_i) and transition probability distribution $P(y_t|y_{t-1})$ that gives the probability of current label based on previous state's label, for $y \in \mathcal{Y}$. A brute-force method would be to compute the probability for all possible sequences $P(x_1, x_2, \ldots, x_T, y_1, y_2, \ldots, y_T)$ and select the sequence having the maximum value. But there are $|\mathcal{Y}|^T$ number of possible sequences, which is extremely high. The Viterbi algorithm uses the following equations to recursively compute maximum probability:

$$V_{1,k} = P(x_1|k) \cdot \pi_k \tag{2.24}$$

$$V_{t,k} = \max_{y \in \mathcal{Y}} \left(P(x_t|k) \cdot P(k|y_{t-1}) \cdot V_{t-1,y} \right), \ \forall k \in \mathcal{Y} \tag{2.25}$$

Here $V_{t,k}$ is the probability of sequence $y_1, y_2, \ldots, y_t$ and π_k is the probability of initial state. The resulting complexity is $|\mathcal{Y}| \times T$ which is much lower than $|\mathcal{Y}|^T$. While finding the maximum probability in Equation 2.25, the y for maximum value is saved in $\mathsf{Ptr}(t, k)$, then following equation is used to retrieve the maximum probable sequence:

$$
\begin{aligned}
y_T &= \operatorname*{argmax}_k V_{T,k},\ \forall k \in \mathcal{Y} \qquad (2.26)\\
y_{t-1} &= \mathsf{Ptr}(t, k),\ t > 1
\end{aligned}
$$

In Equation 2.25, if we use log-probabilities instead of probabilities, then the multiplicative formula turns into an additive one. We can then replace the log-probability with any suitable measure that represents similar properties of relations among variables. We use these versions of the Viterbi algorithm in Section 3.2.2 and 3.2.4.

CHAPTER 3

THEORY AND ALGORITHMS

(Sections of this chapter were published in Asif, Kaiser, Wei Xing, Sima Behpour, and Brian D. Ziebart. "Adversarial Cost-Sensitive Classification." In *UAI*, pp. 92-101. 2015. (Asif et al., 2015), and in Jia Li, Kaiser Asif, Hong Wang, Brian D. Ziebart, and Tanya Y. Berger-Wolf. "Adversarial Sequence Tagging." In *IJCAI*, pp. 1690-1696. 2016. (Li* et al., 2016))

In this chapter we start with notation and formulation of the adversarial framework for cost-sensitive predictions, then we describe the algorithm for solving it, discuss the solution space with synthetic data on a two-dimensional space, and in the end extend the framework for sequence tagging.

3.1 Adversarial Cost-sensitive Classification

The adversarial classifier is a probabilistic classifier that learns a probability distribution that minimizes expected loss of the prediction with respect to the worst case approximation of the evaluation distribution (since the true evaluation distribution is unknown).

3.1.1 Formulation

Let $\hat{\mathsf{P}}(\mathsf{y}|\mathsf{x})$ be the conditional label distribution that the predictor provides, and let the actual evaluation distribution be $\mathsf{P}(\mathsf{y}|\mathsf{x})$. We compactly represent each as a $\mathcal{Y}$-sized vectors using $\hat{\mathbf{p}}_x$ and $\mathbf{p}_x$, $x \in \mathcal{X}$, where $\mathbf{p}_x = [\mathsf{P}(y = 1|x)\ \mathsf{P}(y = 2|x)\ \ldots]^{\mathrm{T}}$. The expected loss suffered from this estimator on input x for a confusion cost matrix $\mathbf{C}$ is:

$$\hat{\mathbf{p}}_{\mathbf{x}}^{\mathrm{T}} \mathbf{C} \mathbf{p}_{\mathbf{x}} = \mathbb{E}_{\hat{\mathsf{P}}(\hat{y}|\mathbf{x})\mathsf{P}(y|\mathbf{x})}[\mathsf{C}_{\hat{Y},Y}].$$

Only samples from the true conditional label distribution $\mathsf{P}(y|x)$ are available. We denote this by distribution $\tilde{\mathsf{P}}(y|x)$ (compactly represented as $\tilde{\mathbf{p}}_x$) and also input sample distribution $\tilde{\mathsf{P}}(x)$. Minimizing the empirical risk under this distribution,

$$\mathbb{E}[\hat{\mathbf{p}}_{\theta,\mathbf{X}} \mathbf{C} \tilde{\mathbf{p}}_{\mathbf{X}}] = \frac{1}{n} \sum_{i=1}^{n} \sum_{\hat{y} \in \mathcal{Y}} \hat{\mathsf{P}}(\hat{y}|\mathbf{x}_i) \mathsf{C}_{\hat{y},y_i},$$

for some parametric form of the estimation distribution, e.g., $\hat{\mathsf{P}}_\theta(y|\mathbf{x}) \propto e^{\theta \cdot \phi(\mathbf{x},y)}$, leads to a non-convex and generally intractable optimization problem, assuming $\mathbf{P} \neq \mathbf{NP}$, as discussed in Section 2.1.

To avoid these non-convex optimization concerns, we employ a robust minimax formulation (Topsøe, 1979; Grünwald and Dawid, 2004) to construct our cost-sensitive classifier (Definition 4). This formulation views the estimation task as a two-player game between an estimator seeking to minimize loss and an adversary seeking to maximize loss. The adversary is constrained to choose distributions that match a vector of moment statistics of the distribution, $\mathbb{E}_{\mathsf{P}(\mathbf{x})\mathsf{P}(y|\mathbf{x})}[\phi(\mathbf{X}, \mathsf{Y})]$. We denote the set of conditional distributions $\mathsf{P}(y|\mathbf{x})$ satisfying these statistics as Ξ.

Definition 4. *In the **constrained cost-sensitive minimax game**, the estimator player first selects a predictive distribution, $\hat{\mathbf{p}}_{\mathbf{x}} \triangleq \hat{P}(\hat{y}|\mathbf{x}) \in \Delta$, for each input $\mathbf{x}$, from the conditional probability simplex Δ, and then the adversarial player selects an evaluation distribution, $\check{\mathbf{p}}_{\mathbf{x}} \triangleq \check{P}(\check{y}|\mathbf{x}) \in \Delta$, for each input x from the set Ξ of distributions consistent with known statistics:*

$$\min_{\{\hat{\mathbf{p}}_{\mathbf{x}}\}\in\boldsymbol{\Delta}} \max_{\{\check{\mathbf{p}}_{\mathbf{x}}\}\in\Xi\cap\boldsymbol{\Delta}} \mathbb{E}_{P(\mathbf{x})}[\hat{\mathbf{p}}_{\mathbf{X}}^{T}\mathbf{C}\check{\mathbf{p}}_{\mathbf{X}}] \tag{3.1}$$

$$\text{where: } \Xi : \mathbb{E}_{P(\mathbf{x})\check{P}(\check{y}|\mathbf{x})}[\phi(\mathbf{X},\check{Y})] = \tilde{\phi}.$$

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

Conceptually, the feature statistics $\phi(\mathbf{x}, y)$ defining the set Ξ should be chosen to restrict the adversary as much as possible from maximizing the loss. However, defining the set to be too restrictive leads to overfitting to the training data. Indeed, the complexity of the estimator $\hat{P}(\hat{y}|\mathbf{x})$ implicitly grows with the dimensionality of the constraints in Ξ. Thoughtfully specifying the feature function $\phi(\cdot,\cdot)$ and employing regularization can avoid this issue (section 3.1.4).

3.1.2 Inference As Zero-sum Game Equilibrium

We establish efficient inference algorithms for our approach in this section. Theorem 2 transforms the joint adversary-constrained zero-sum games over many different inputs $\mathbf{x}$ into a set of unconstrained zero-sum game that are independent for each input $\mathbf{x}$ and connected by a parameterized cost matrix defining each player's game outcomes.

Theorem 2. *Determining the value of the constrained cost-sensitive minimax game reduces to a minimization over the expectation of many unconstrained minimax game:*

$$
\begin{aligned}
&\min_{\{\hat{\mathbf{p}}_{\mathbf{x}}\}\in\boldsymbol{\Delta}} \max_{\{\check{\mathbf{p}}_{\mathbf{x}}\}\in\Xi\cap\boldsymbol{\Delta}} \mathbb{E}_{P(\mathbf{x})}[\hat{\mathbf{p}}_{\mathbf{X}}^{\mathrm{T}}\mathbf{C}\check{\mathbf{p}}_{\mathbf{X}}] && (3.2)\\
&= \max_{\{\check{\mathbf{p}}_{\mathbf{X}}\}\in\Xi\cap\boldsymbol{\Delta}} \mathbb{E}_{P(\mathbf{x})}\left[\min_{\hat{\mathbf{p}}_{\mathbf{X}}\in\Delta} \hat{\mathbf{p}}_{\mathbf{X}}^{\mathrm{T}}\mathbf{C}\check{\mathbf{p}}_{\mathbf{X}}\right] && (3.3)\\
&= \min_{\theta} \mathbb{E}_{P(\mathbf{x})}\left[\max_{\check{\mathbf{p}}_{\mathbf{x}}\in\Delta} \min_{\hat{\mathbf{p}}_{\mathbf{X}}\in\Delta} \hat{\mathbf{p}}_{\mathbf{X}}^{\mathrm{T}}\mathbf{C}'_{\mathbf{X},\theta}\check{\mathbf{p}}_{\mathbf{X}}\right], && (3.4)
\end{aligned}
$$

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

Proof of Theorem 2.

$$
\begin{aligned}
&\min_{\{\hat{\mathbf{p}}_{\mathbf{x}}\}\in\boldsymbol{\Delta}} \max_{\{\check{\mathbf{p}}_{\mathbf{x}}\}\in\Xi\cap\boldsymbol{\Delta}} \mathbb{E}_{P(\mathbf{x})}[\hat{\mathbf{p}}_{\mathbf{X}}^{\mathrm{T}}\mathbf{C}\check{\mathbf{p}}_{\mathbf{X}}]\\
&\overset{(a)}{=} \max_{\{\check{\mathbf{p}}_{\mathbf{x}}\}\in\Xi\cap\boldsymbol{\Delta}} \min_{\{\hat{\mathbf{p}}_{\mathbf{x}}\}\in\boldsymbol{\Delta}} \mathbb{E}_{P(x)}[\hat{\mathbf{p}}_{\mathbf{X}}^{\mathrm{T}}\mathbf{C}\check{\mathbf{p}}_{\mathbf{X}}]\\
&\overset{(b)}{=} \max_{\{\check{\mathbf{p}}_{\mathbf{x}}\}\in\Xi\cap\boldsymbol{\Delta}} \mathbb{E}_{P(\mathbf{x})}\left[\min_{\hat{\mathbf{p}}_{\mathbf{X}}\in\Delta} \hat{\mathbf{p}}_{\mathbf{X}}^{\mathrm{T}}\mathbf{C}\check{\mathbf{p}}_{\mathbf{X}}\right]\\
&\overset{(c)}{=} \max_{\{\check{\mathbf{p}}_{\mathbf{x}}\}\in\boldsymbol{\Delta}} \min_{\theta} \mathbb{E}_{P(\mathbf{x})}\left[\min_{\hat{\mathbf{p}}_{\mathbf{X}}\in\Delta} \hat{\mathbf{p}}_{\mathbf{X}}^{\mathrm{T}}\mathbf{C}\check{\mathbf{p}}_{\mathbf{X}}\right]\\
&\qquad\qquad + \theta^{\mathrm{T}}\mathbb{E}_{P(\mathbf{x})}[\Phi_{\mathbf{X}}\left(\check{\mathbf{p}}_{\mathbf{X}} - \tilde{\mathbf{p}}_{\mathbf{X}}\right)]\\
&\overset{(d)}{=} \min_{\theta} \mathbb{E}_{P(\mathbf{x})}\left[\max_{\check{\mathbf{p}}_{\mathbf{X}}\in\Delta} \min_{\hat{\mathbf{p}}_{\mathbf{X}}\in\Delta} \hat{\mathbf{p}}_{\mathbf{X}}^{\mathrm{T}}\mathbf{C}'_{\mathbf{X},\theta}\check{\mathbf{p}}_{\mathbf{X}}\right]
\end{aligned}
$$

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

$$(C'_{\mathbf{x}})_{\hat{y},\check{y}} = C_{\hat{y},\check{y}} + \theta^T(\phi(\mathbf{x}, \check{y}) - \phi(\mathbf{x}, \tilde{y})). \tag{3.5}$$

Step (a) follows from minimax duality in zero-sum games (von Neumann and Morgenstern, 1947), discussed in section 2.5. As an affine function of terms each with individual $\check{\mathbf{p}}_{\mathbf{x}}$ term, each minimization can be performed independently in step (b). Step (c) expresses the primal Lagrangian. For step (d), $\mathbb{E}_{P(\mathbf{x})}[\min_{\hat{\mathbf{p}}_{\mathbf{X}} \in \Delta}\ \hat{\mathbf{p}}_{\mathbf{X}}^T \mathbf{C} \check{\mathbf{p}}_{\mathbf{X}} + \theta^T \Phi_{\mathbf{X}}(\check{\mathbf{p}}_{\mathbf{X}} - \tilde{\mathbf{p}}_{\mathbf{X}})]$—a non-negative linear combination of minimums of affine functions—is a concave function of $\check{\mathbf{p}}_{\mathbf{x}}$ terms and an affine (hence convex) function of θ. The domain of $\check{\mathbf{p}}_{\mathbf{x}}$ is closed and convex and domain of θ is convex, strong duality holds (Sion, 1958), (section 2.6). Finally, as in step (b), the maximizations can then be independently applied. □

Figure 2 shows the value of the game for a single $\mathbf{x}$ from Equation 3.3 as a function of the adversarial distribution $\check{\mathbf{p}}_{\mathbf{x}}$ for zero-one loss and a more general cost matrix. The adversary is not free to independently maximize these functions for each $\mathbf{x}$, but must instead choose a structured prediction that resides within the constraint set Ξ.

After applying Theorem 2 and given model parameters, θ, (obtaining these parameters is discussed in section 3.1.3) the unconstrained game, $\max_{\check{\mathbf{p}}_{\mathbf{x}} \in \Delta}\ \min_{\hat{\mathbf{p}}_{\mathbf{x}} \in \Delta} \hat{\mathbf{p}}_{\mathbf{x}}^T \mathbf{C}'_{\mathbf{x},\theta} \check{\mathbf{p}}_{\mathbf{x}}$, can be

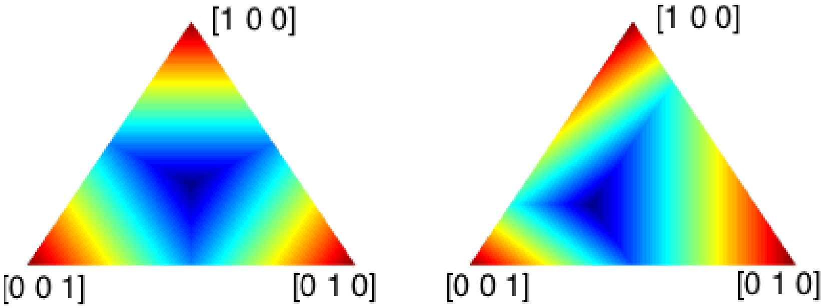


Figure 2: The portion of the adversary's objective function Equation (Equation 3.3) for a single example, $\min_{\hat{\mathbf{p}}_{\mathbf{x}} \in \Delta} \hat{\mathbf{p}}_{\mathbf{x}}^{\mathrm{T}} \mathbf{C} \check{\mathbf{p}}_{\mathbf{x}}$, in the adversary-constrained game for zero-one loss (left) and a more general cost-sensitive loss with cost matrix [0 2 3; 2 0 1; 1 3 0] (right) in a three-class prediction task.

solved independently for each $\mathbf{x}$. In this augmented game, our original cost matrix from Eq. Equation (Equation 1.1) is transformed into the augmented cost matrix:

$$\mathbf{C}' = \begin{bmatrix} 0+\psi_1 & 1+\psi_2 & 2+\psi_3 & 0+\psi_4 \\ 3+\psi_1 & 0+\psi_2 & 1+\psi_3 & 3+\psi_4 \\ 4+\psi_1 & 2+\psi_2 & 0+\psi_3 & 1+\psi_4 \\ 1+\psi_1 & 1+\psi_2 & 2+\psi_3 & 0+\psi_4 \end{bmatrix}, \tag{3.6}$$

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

Figure 3 shows the adversary's objective function in the unconstrained, cost-augmented game. Conceptually, the adversary's objective function from the constrained game (Figure 2) is "placed" on top of a hyperplane shaped by the Lagrangian potential terms, ψ_i. The difference in these potential terms determines the adversary's equilibrium strategy. For the binary classification task, there are three possible equilibrium strategies for the adversary, two pure strategies for the two classes and one strategy that is the mixture of the two. With three classes, there are seven possibilities:w three pure strategies; three strategies that are mixtures of two classes; and one strategy that is a mixture of all three classes.

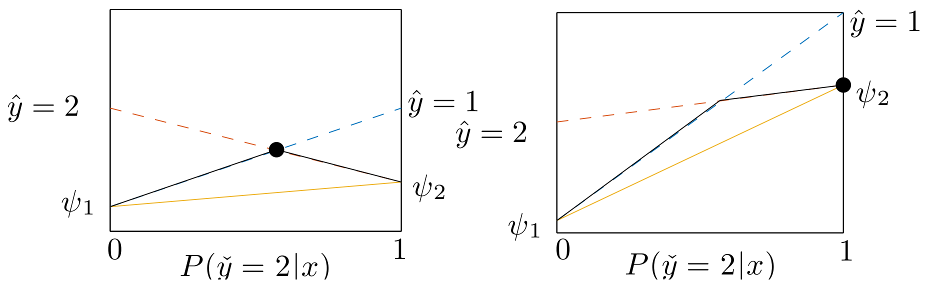


Figure 3: The adversary's objective in the unconstrained game for a binary classification task with a mixed (uncertain) equilibrium solution (left) and a pure (certain) equilibrium solution (right). The third adversary strategy, $P(\check{y} = 2|\mathbf{x}) = 0$, is realized when $\psi_1 >> \psi_2$.

Unlike the logarithmic loss under this minimax formulation, which yields members of the exponential family (Wainwright and Jordan, 2008), the cost-sensitive loss function does not generally provide a closed-form parametric solution. Instead, the inner minimax game (inside the expectation of Equation 3.4) for each input $\mathbf{x}$ can be solved as a linear program (von Neumann and Morgenstern, 1947), discussed in Section 2.5:

$$\begin{aligned} &\max_{\nu, \check{P}(\check{y}|\mathbf{x})} \nu \qquad (3.7)\\ \text{subject to: } &\nu \leq \sum_{\check{y} \in \mathcal{Y}} \check{P}(\check{y}|\mathbf{x}) (C'_{\mathbf{x},\theta})_{\hat{y},\check{y}} \; \forall \hat{y} \in \mathcal{Y} \\ &\sum_{\check{y} \in \mathcal{Y}} \check{P}(\check{y}|\mathbf{x}) = 1 \text{ and } \check{P}(\check{y}|\mathbf{x}) \geq 0, \; \forall \check{y} \in \mathcal{Y}. \end{aligned}$$

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

3.1.3 Learning via Convex Optimization

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

Theorem 3. *The subdifferential of the outer minimization problem (Equation 3.4) includes the expected feature difference as a subgradient:*

$$\begin{aligned}&\mathbb{E}_{\mathsf{P}(\mathbf{x})\check{\mathsf{P}}^*_{\hat{\theta}}(\check{y}|\mathbf{x})}\left[\phi(\mathbf{X},\check{Y})\right] - \mathbb{E}_{\mathsf{P}(\mathbf{x})\mathsf{P}(y|\mathbf{x})}\left[\phi(\mathbf{X},Y)\right] \\ &\qquad \in \partial_\theta \mathbb{E}_{\mathsf{P}(\mathbf{x})}\left[\min_{\hat{\mathbf{p}}_\mathbf{x}\in\Delta}\max_{\check{\mathbf{p}}_\mathbf{x}\in\Delta} \hat{\mathbf{p}}_\mathbf{X}^{\mathrm{T}}\mathbf{C}'_{\mathbf{X},\theta}\check{\mathbf{p}}_\mathbf{X}\right]\Bigg|_{\theta=\hat{\theta}}\end{aligned} \tag{3.8}$$

where $\check{\mathsf{P}}^*(\check{y}|\mathbf{x})$ *is the solution to Equation 3.7.*

Proof of Theorem 3. Taking the subdifferential, we have:

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

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

Leveraging the convexity of the formulation's objective function (discussed in the Proof of Theorem 2), and using the common substitution of the sample training data distribution, $\tilde{P}(\mathbf{x})$, in place of the distribution $P(\mathbf{x})$, we employ standard subgradient-based optimization methods for convex optimization problems to obtain parameters for our cost-sensitive classifier (Algorithm 1).

Algorithm 1 Parameter estimation for the robust cost-sensitive classifier

```
Require: Cost matrix $\mathbf{C}$, training dataset $\mathcal{D}$ with pairs $(\tilde{\mathbf{x}}_i, \tilde{y}_i) \in \mathcal{D}$, feature function $\phi : \boldsymbol{\mathcal{X}} \times \mathcal{Y} \to \mathbb{R}^k$, time-varying learning rate $\{\gamma_t\}$
Ensure: Model parameter estimate $\theta$
  $t \leftarrow 1$
  while $\theta$ not converged do
    for all $(\tilde{\mathbf{x}}_i, \tilde{y}_i) \in \mathcal{D}$ do
      Construct cost matrix $\mathbf{C}'_{\tilde{\mathbf{x}}_i,\theta}$ using Equation 3.5
      Solve for $\check{P}(\check{y}|\tilde{\mathbf{x}}_i)$ using the LP of Equation 3.7
      $\nabla_\theta = \mathbb{E}_{\check{P}(\check{y}|\tilde{\mathbf{x}}_i)}[\phi(\tilde{\mathbf{x}}_i, \check{Y})] - \phi(\tilde{\mathbf{x}}_i, \tilde{y}_i)$
      $\theta = \theta - \gamma_t \nabla_\theta$
      $t \leftarrow t + 1$
    end for
  end while
```

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

3.1.4 Performance Guarantees & Illustrative Examples

We establish performance guarantees and illustrate the behavior of our approach in this section. We focus specifically on the similarities to and differences from support vector machines (Cortes and Vapnik, 1995) and their structured extensions (Tsochantaridis et al., 2004). Given ideal data (linearly separable), Theorem 4 establishes an equivalence to hard-margin SVMs.

Theorem 4. *Given linearly separable training data, i.e.,*

$$\exists \theta : \forall i, y' \neq y_i, \theta \cdot \phi(\mathbf{x}_i, y_i) > \theta \cdot \phi(\mathbf{x}_i, y'), \tag{3.9}$$

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

Proof. Equation 3.9 implies $\exists \theta' : \forall i, y' \neq y_i, \theta' \cdot \phi(\mathbf{x}_i, y_i) > \theta' \cdot \phi(\mathbf{x}_i, y') + C_{y',y_i}$ (the hard-margin cost-sensitive SVM constraint set with $\epsilon = \mathbf{0}$ in Equation 2.7) by multiplicatively scaling θ. The Nash equilibrium is $\check{P}(\check{y}_i|\mathbf{x}_i) = 1$ and $\hat{P}(\hat{y}_i|\mathbf{x}_i) = 1$ with a cost-sensitive loss of zero *if and only if* this inequality is satisfied. Given this, the dual optimization in Equation 3.4 realizes its minima (zero loss) only when these constraints are satisfied. The L_2 regularization term is a monotonic transformation of the objective of the hard-margin SVM: $\theta \cdot \theta$. Thus, having the same constraints and objective functions with corresponding maxima, an equivalent solution is produced. □

As a result of this equivalence to hard-margin SVM, adversarial classification inherits the convergence properties of support vectors machines in the realizable case of Equation 3.9.

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

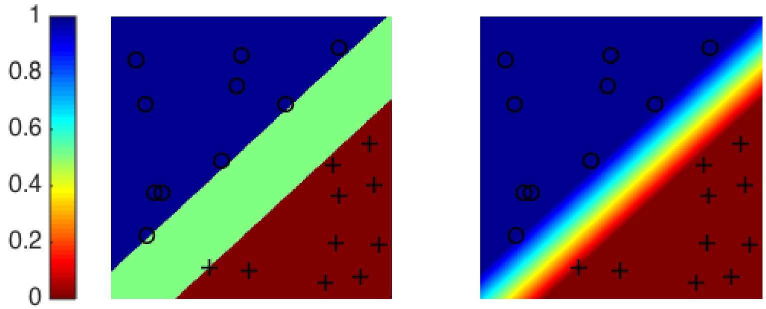


Figure 4: Adversary (left) and predictor (right) distributions for separable data under zero-one loss

Given non-separable data, the adversarial approach suggests choosing a set Ξ of constraints based on training samples $\tilde{P}(\mathbf{x}, y)$ that will also contain the true label distribution, $P(y|\mathbf{x})$. When this is accomplished, Theorem 5 provides performance guarantees for generalization.

Theorem 5. *If* $\mathsf{P}(\mathsf{y}|\mathbf{x}) \in \Xi$*, confusion costs from the adversarial game upper bound the generalization error confusion costs:*

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

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

Slack can be added to the constraint set Ξ or regularization to the dual optimization problem of Equation 3.3 to address finite sample approximation error when using sample data, $\mathbb{E}_{\tilde{\mathsf{P}}(\mathbf{x},\mathsf{y})}[\phi(\mathbf{X},\mathsf{Y})]$, as an estimate of the distribution's statistics, $\mathbb{E}_{\mathsf{P}(\mathbf{x},\mathsf{y})}[\phi(\mathbf{X},\mathsf{Y})]$.

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

The equilibria under cost-sensitive losses, shown in Figure 6 shifts the region of uncertainty to better minimize the expected cost compared to Figure 5, which is based on the same data

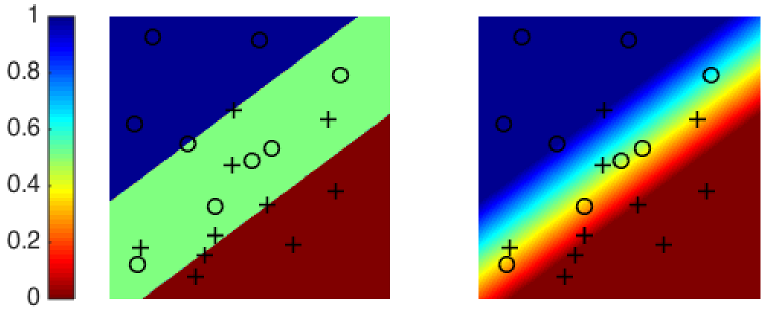


Figure 5: Adversary (left) and predictor (right) distributions for nonseparable data under zero-one loss

sample. Additionally, the adversary's predictions shift ($\check{\mathsf{P}}(\mathsf{Y} = \text{‘o’}|\mathbf{x}) = .25$) within the region of uncertainty.

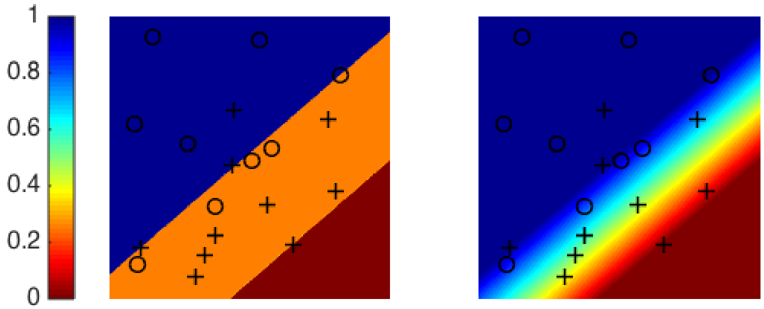


Figure 6: Adversary (left) and predictor (right) distributions for nonseparable data under [0 1; 3 0] cost matrix.

From the perspective of Theorem 4 and Theorem 5, adversarial cost-sensitive classification provides an alternative to hinge-loss "softening" of the hard-margin SVM. By posing cost-sensitive prediction as an adversarial game (Definition 4), our approach approximates aspects of the training data while being able to employ non-convex loss functions without the intractability encountered by empirical risk minimization. Prediction under this approach reduces to the well-studied problem of solving a zero-sum game, which is easily addressed using linear programming via Equation 3.7. This is only a little more complicated than predictions for SVM based on the label that maximizes a linear potential function. Like SVMs, estimating model parameters can be posed as a convex optimization problem and solved using subgradient optimization methods (Algorithm 1) under our approach.

3.1.5 Fisher Consistency

The premise of Fisher Consistency assumes that we have rich feature representation and true evaluation distribution is available (Definition 1). In that case adversary's distribution equals the true distribution as the subgradient is zero at the optima, the constraint becomes zero, and the predictor finds a distribution that minimizes the cost

$$\min_{\hat{\mathbf{p}}_{\mathbf{x}} \in \Delta} \hat{\mathbf{p}}_{\mathbf{X}}^{\mathrm{T}} \mathbf{C} \mathbf{p}_{\mathbf{X}}, \tag{3.10}$$

where $\mathbf{p}_{\mathbf{X}}$ is the true evaluation distribution. By definition of Fisher consistency, Equation 3.10 minimizes the loss on the true evaluation (population) distribution and hence adversarial classification method is Fisher consistent.

3.1.6 Potential Based Prediction

As shown in Figure 3, the maximum potential value aligns with the deterministic class. Further (Fathony et al., 2018) show that potential based prediction is consistent as well when correct prediction has strictly smaller cost. Therefore we can use potential-based prediction like ERM-based models, e.g. SVM, when a probabilistic prediction is not required:

$$y* = \operatorname*{argmax}_{y} \theta\phi(\mathbf{x}, y)$$

On the other hand, in section 3.2.4, although the parameter optimization is faster, but we sacrifice the ability to retrieve the predictor's distribution from the formulation. We use potential based prediction is such case.

Figure 7 shows that potential based classification, i.e. assigning class label to maximum potential matches the probability based prediction. From the bottom plots we can see that potentials for class 2 on the right has lower values than corresponding potentials for class 1 on the left image along a line that is below the diagonal of the plot area, which matches the decision boundary of the predictor plot (top right in the figure).

3.2 Adversarial Sequence Tagging

In this section, we extend the adversarial prediction framework to sequence tagging tasks. In a sequence tagging task, there is a sequence of nodes or variables and the task is to predict each node's target class based on its feature and also adjacent nodes' classes. The loss is the sum of misclassification cost of all variables of the sequence. To address sequence tagging,

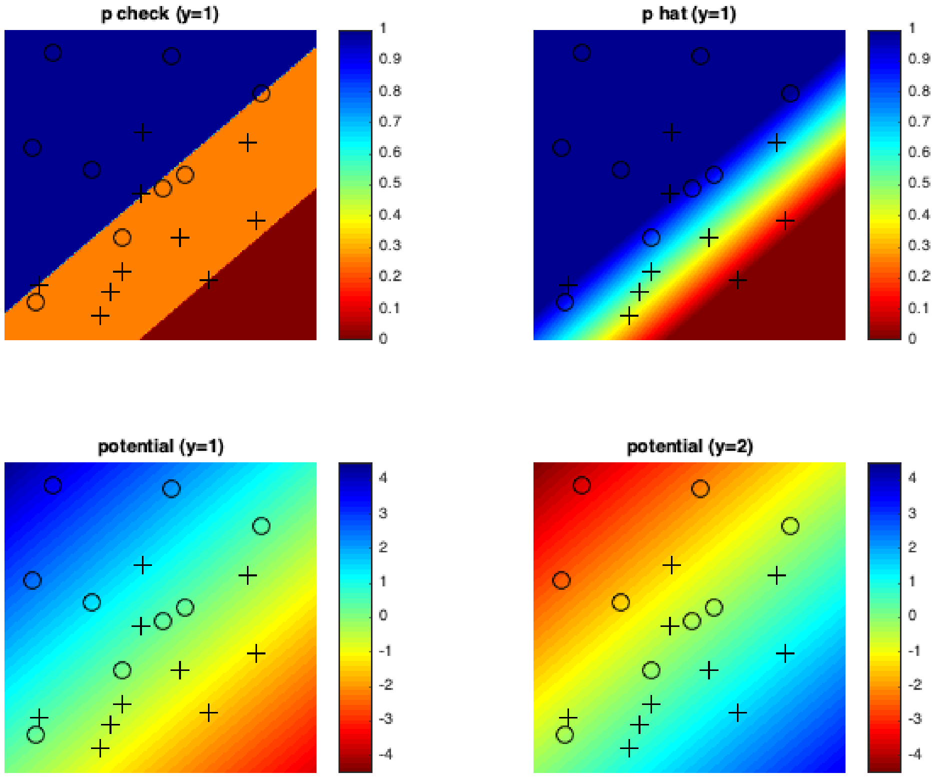


Figure 7: Feature potentials corresponding to Figure 6. Top row repeats Figure 6. In the bottom, left displays the potential for class 1, right for class 2.

first, we formulate the zero-sum game using the additive misclassification loss, where joint probabilities for the full sequence are learned using double oracle method. Then utilizing the additive decomposability, we modify the objectives to improve the computation efficiency using a method called single oracle (Li* et al., 2016). Finally, we note that Markov independence property allows us to consider only pairwise probabilities of adjacent nodes instead of the joint probabilities of the whole sequence, and this enables a more efficient optimization.

3.2.1 Formulation

In order to predict a sequence of predictions, the predictor chooses a conditional probability distribution, $\hat{P}(\hat{\mathbf{y}}|\mathbf{x})$, against the adversary's distribution $\check{P}(\check{\mathbf{y}}|\mathbf{x})$. The adversary is constrained to match the feature statistics $\Phi(\mathbf{x}, \mathbf{y})$. The optimization problem is similar to the single-variate problem where the goal of estimator is to minimize the expected loss while adversary's goal is to maximize the loss:

$$\min_{\hat{P}(\hat{\mathbf{y}}|\mathbf{x})} \max_{\check{P}(\check{\mathbf{y}}|\mathbf{x})} \mathbb{E}_{\tilde{P}(\mathbf{x})\check{P}(\check{\mathbf{y}}|\mathbf{x})\hat{P}(\hat{\mathbf{y}}|\mathbf{x})} \left[\sum_{t=1}^{T} C_{\hat{Y}_t, \check{Y}_t} \right] \tag{3.11}$$
$$\text{such that: } \mathbb{E}_{\tilde{P}(\mathbf{x})\check{P}(\check{\mathbf{y}}|\mathbf{x})}[\Phi(\mathbf{X}, \check{\mathbf{Y}})] = \mathbb{E}_{\tilde{P}(\mathbf{x},\mathbf{y})}[\Phi(\mathbf{X}, \mathbf{Y})],$$

where the feature functions, $\Phi(\mathbf{x}, \mathbf{y})$, can be decomposed over pairs of the $Y_1, \dots, Y_T$ variables: e.g., $\Phi(\mathbf{x}, \mathbf{y}) = \sum_{t=1}^{T-1} \phi(\mathbf{x}, y_t, y_{t+1})$.

Using Lagrangian and zero-sum game duality Equation 3.11 reduces to a convex optimization problem (Theorem 2):

$$\min_{\theta} \mathbb{E}_{\tilde{P}(\tilde{x},\tilde{y})} \left[\max_{\check{\mathbf{p}}_{\mathbf{X}}} \min_{\hat{\mathbf{p}}_{\mathbf{X}}} \hat{\mathbf{p}}_{\mathbf{X}}^{T} \mathbf{C}'_{\mathbf{X},\theta} \check{\mathbf{p}}_{\mathbf{X}} \right], \tag{3.12}$$

where $\hat{\mathbf{p}}_{\mathbf{X}} = \{\hat{P}(\hat{\mathbf{y}}|\mathbf{x})\}$ and $\check{\mathbf{p}}_{\mathbf{X}} = \{\check{P}(\check{\mathbf{y}}|\mathbf{x})\}$ are vector representations of the conditional probabilities, and $\mathbf{C}'_{X,\theta}$ is the potential-augmented cost-matrix, payoff matrix for the zero-sum game, where each cell represents the total loss of the sequence mismatch between the adversary and the predictor plus the Lagrangian potential term that enforces the optimization's constraints using the Lagrangian parameter θ: $(\mathbf{C}'_{\mathbf{x},\mathbf{y},\theta})_{\hat{\mathbf{y}},\check{\mathbf{y}}} = \text{loss}(\hat{\mathbf{y}},\check{\mathbf{y}}) + \theta \cdot (\Phi(\mathbf{x},\check{\mathbf{y}}) - \Phi(\mathbf{x},\mathbf{y})) = \sum_{t=1}^{T} C_{\hat{y}_t,\check{y}_t} + \theta \cdot (\Phi(\mathbf{x},\check{\mathbf{y}}) - \Phi(\mathbf{x},\mathbf{y}))$.

Considering 0-1 loss for each node is a special case of cost-sensitive sequence tagging. This is equivalent to the number of misclassifications for a sequence, and is called the Hamming loss. For example if predicted and true sequences are 001 and 101, then the Hamming loss is 1. We consider this cost-insensitive loss in (Li* et al., 2016). This model can easily be extended to a cost-sensitive model. Table II shows a 3-length binary-valued (target classes are either 0 or 1) sequence game's payoff matrix which has the Hamming loss and a Lagrangian potential for each cell that corresponds to the predictor and adversary's choices.

As with the classification problem, these zero-sum games can be solved using linear programming to find each player's mixed Nash equilibrium (von Neumann and Morgenstern, 1947), where a pure strategy is an assignment of labels to the full sequence. The mixed Nash equilibrium strategy for the adversarial player can be obtained from:

$$\max_{\check{\mathbf{p}} \geq \mathbf{0}, v} \quad v \tag{3.13}$$

$$\text{such that: } v \leq \mathbf{C}'_{\hat{\mathbf{y}},*}\check{\mathbf{p}} \;\forall \hat{\mathbf{y}} \in \boldsymbol{\mathcal{Y}}$$

$$\mathbf{1}^{\mathrm{T}}\check{\mathbf{p}} = 1.$$

TABLE II: The payoff matrix $\mathbf{C}'_{x,\theta}$ for a game over the length three binary-valued chain of variables between player $\check{Y}$ choosing a distribution over columns and $\hat{Y}$ choosing a distribution over rows. Lagrangian potentials are compactly represented as: $\psi_{\check{y}_1\check{y}_2\check{y}_3} \triangleq \theta \cdot (\Phi(\check{\mathbf{y}},\mathbf{x}) - \Phi(\mathbf{y},\mathbf{x}))$.

	000	**001**	**010**	**011**	**100**	**101**	**110**	**111**
000	$0+\psi_{000}$	$1+\psi_{001}$	$1+\psi_{010}$	$2+\psi_{011}$	$1+\psi_{100}$	$2+\psi_{101}$	$2+\psi_{110}$	$3+\psi_{111}$
001	$1+\psi_{000}$	$0+\psi_{001}$	$2+\psi_{010}$	$1+\psi_{011}$	$2+\psi_{100}$	$1+\psi_{101}$	$3+\psi_{110}$	$2+\psi_{111}$
010	$1+\psi_{000}$	$2+\psi_{001}$	$0+\psi_{010}$	$1+\psi_{011}$	$2+\psi_{100}$	$3+\psi_{101}$	$1+\psi_{110}$	$2+\psi_{111}$
011	$2+\psi_{000}$	$1+\psi_{001}$	$1+\psi_{010}$	$0+\psi_{011}$	$3+\psi_{100}$	$2+\psi_{101}$	$2+\psi_{110}$	$1+\psi_{111}$
100	$1+\psi_{000}$	$2+\psi_{001}$	$2+\psi_{010}$	$3+\psi_{011}$	$0+\psi_{100}$	$1+\psi_{101}$	$1+\psi_{110}$	$2+\psi_{111}$
101	$2+\psi_{000}$	$1+\psi_{001}$	$3+\psi_{010}$	$2+\psi_{011}$	$1+\psi_{100}$	$0+\psi_{101}$	$2+\psi_{110}$	$1+\psi_{111}$
110	$2+\psi_{000}$	$3+\psi_{001}$	$1+\psi_{010}$	$2+\psi_{011}$	$1+\psi_{100}$	$2+\psi_{101}$	$0+\psi_{110}$	$1+\psi_{111}$
111	$3+\psi_{000}$	$2+\psi_{001}$	$2+\psi_{010}$	$1+\psi_{011}$	$2+\psi_{100}$	$1+\psi_{101}$	$1+\psi_{110}$	$0+\psi_{111}$

Similarly, the predictor's optimal mixed strategy is:

$$\begin{aligned} &\min_{\hat{\mathbf{p}}\geq\mathbf{0},\nu} \quad \nu \\ &\text{such that: } \nu \geq \hat{\mathbf{p}}^{\mathrm{T}}\mathbf{C}'_{*,\check{y}} \; \forall \check{\mathbf{y}} \in \boldsymbol{\mathcal{Y}} \\ &\mathbf{1}^{\mathrm{T}}\hat{\mathbf{p}} = 1. \end{aligned} \tag{3.14}$$

However, solving these matrix games directly using the method of adversarial classification (Asif et al., 2015) becomes intractable as for each player we now have $|\mathcal{Y}|^{\mathsf{T}}$ choices in the game matrix $\mathbf{C}'_{\mathbf{x},\theta}$, since there are $|\mathcal{Y}|$ possible assignments of labels for each of the T variables. The size of the payoff matrix becomes $|\mathcal{Y}|^{2\mathsf{T}}$. To address this intractability several approaches are addressed below.

3.2.2 Double Oracle Method for Efficient Prediction

To reduce the computational cost of solving the entire adversarial game, we use the double oracle algorithm (McMahan et al., 2003). It constructs the game matrix iteratively until finding the correct equilibrium. First, a subset of pure strategies are chosen, $\hat{S}$ and $\check{S}$ for the predictor and adversary player respectively, using these strategy sets, which are the label assignments of the sequence, the payoff matrix similar to Table II is constructed, then Equation 3.13 or Equation 3.14 are used to get the probability distribution of the corresponding mixed strategies. Using the mixed strategies of the equilibrium, $\hat{P}(\hat{\mathbf{y}}|\mathbf{x})$ or $\check{P}(\check{\mathbf{y}}|\mathbf{x})$, it then finds the best response pure strategy for the other player $\check{\mathbf{y}}_{BR}$ or $\hat{\mathbf{y}}_{BR}$ respectively and added to the associated strategy sets $\check{S}$ or $\hat{S}$. The algorithm terminates when neither of the players can improve by adding anymore best responses, i.e. adding a best response predictor player cannot reduce the game value or adversary player cannot increase the game value by adding its best response. The best response pure strategy $\check{\mathbf{y}}_{BR}$ is computed using:

$$\begin{aligned}
&\max_{\check{\mathbf{y}}_{1:T}} \mathbb{E}_{\hat{P}(\hat{\mathbf{y}}_{1:T}|\mathbf{x})}\left[\sum_{t=1}^{T} I(\hat{Y}_t \neq \check{y}_t)\right] + \sum_{t=1}^{T-1} \theta \cdot \phi(\mathbf{x}, \check{\mathbf{y}}_{t:t+1}) \\
=&\max_{\check{y}_1}\Bigg(\mathbb{E}_{\hat{P}(\hat{y}_1|\mathbf{x})}\left[I(\hat{Y}_1 \neq \check{y}_1)\right] + \max_{\check{y}_2}\Bigg(\theta \cdot \phi(\mathbf{x}, \check{\mathbf{y}}_{1:2}) \qquad (3.15)\\
&+ \mathbb{E}_{\hat{P}(\hat{y}_2|\mathbf{x})}\left[I(\hat{Y}_2 \neq \check{y}_2)\right] + \max_{\check{y}_3}\Bigg(\theta \cdot \phi(\mathbf{x}, \check{\mathbf{y}}_{2:3}) + \quad \ldots \\
&+ \max_{\check{y}_T} \theta \cdot \phi(\mathbf{x}, \check{\mathbf{y}}_{T-1:T}) + \mathbb{E}_{\hat{P}(\hat{y}_T|\mathbf{x})}\left[I(\hat{Y}_T \neq \check{y}_T)\right]\Bigg)\Bigg)\Bigg),
\end{aligned}$$

which follows the Viterbi algorithm (Viterbi, 1967) that iteratively computes maximum possible loss using previous subsequence and link between current node and the last node of the subsequence. To compute the expected cost via Viterbi, marginal probabilities for the nodes are required, which are computed by $\hat{P}(\hat{Y}_t = y|\mathbf{x}) = \sum_{\hat{\mathbf{y}}_s} \hat{P}(\hat{\mathbf{y}}_s|\mathbf{x}), \forall \hat{\mathbf{y}}_s \in \hat{S}$ where $\hat{\mathbf{y}}_{s,t} = y,\ y \in [0, 1]$. Best response $\hat{\mathbf{y}}_{BR}$ is computed similarly using adversary's $\check{P}(\check{\mathbf{y}}|\mathbf{x})$ distribution but without the feature potential terms and finding the minimum expected loss: $\operatorname{argmin}_{\hat{\mathbf{y}}} \mathbb{E}_{\check{P}(\check{\mathbf{y}}|\mathbf{x})} \left[\sum_{t=1}^{T} I(\hat{y}_t \neq \check{Y}_t)\right]$.

Note that for cost-sensitive sequence tagging this formulation can easily adapted by replacing the 0-1 loss $I(\hat{Y}_t \neq \check{y}_t)$ by the misclassification cost of the label $C_{\hat{Y}_t, \check{Y}_t}$ for each node.

3.2.3 Single Oracle Method for Efficient Prediction

Unlike adversarial prediction methods for structured losses (Wang et al., 2015), the sequence tagging loss can be additively decomposed into payoff matrix terms $\mathbf{C}_t$ for $t \in \{1, ..., T\}$ as can be observed in Equation 3.15. This makes the estimator's predictions independent of each other and all the "pure strategies" (assumed by all the possible labels for each node) can be considered at once efficiently using the following pair of linear programs:

$$(1) \min_{\hat{\mathbf{p}}_1, \hat{\mathbf{p}}_2, ..., \hat{\mathbf{p}}_T, \nu} \nu \text{ such that: } \hat{\mathbf{p}}_t \geq \mathbf{0} \text{ and } \mathbf{1}^T \hat{\mathbf{p}}_t = 1, \forall t; \tag{3.16}$$

$$\text{and } \nu \geq \theta^T \phi(\mathbf{x}, \check{y}) + \sum_{t=1}^{T} \hat{\mathbf{p}}_t^T [\mathbf{C}_t]_{*,\check{y}}\ \check{y} \in \check{S};$$

$$(2) \max_{\check{\mathbf{p}} \geq \mathbf{0}, \nu_1, \nu_2, ..., \nu_T} \theta^T \mathbf{\Phi}_{\mathbf{x},\mathbf{y}} \check{\mathbf{p}} + \sum_{t=1}^{T} \nu_t \text{ such that: } \mathbf{1}^T \check{\mathbf{p}} = 1 \tag{3.17}$$

$$\text{and } \nu_t \leq [\mathbf{C}_t]_{\hat{y},*}\ \check{\mathbf{p}}\ \forall t, \hat{y} \in \mathcal{Y};$$

Here $\hat{\mathbf{p}}_{\mathsf{t}}$ are the predictor's distributions for each individual node, and $\check{\mathbf{p}}$ is the adversary's joint probability distribution for the full sequence assignment. As the entire set of predictor pure strategies is considered via $\hat{\mathsf{y}}_{\mathsf{t}}$ terms, the oracle now only needs to iteratively expand the adversary's set of strategy and becomes single oracle (Algorithm 2).

Algorithm 2 Single Oracle Game Solver

```
Require: Lagrangian potential, $\boldsymbol{\psi}$; initial action set $\check{\mathsf{S}}$
Ensure: $[\hat{\mathsf{P}}(\hat{\mathbf{y}}|\mathbf{x}), \check{\mathsf{P}}(\check{\mathbf{y}}|\mathbf{x})]$
  $\check{\mathbf{y}}_{\mathsf{BR}} \leftarrow \{\}$
  repeat
    $\mathbf{C}_{\mathsf{t}} \leftarrow$ buildPayoffMatrices($\check{\mathsf{S}}$, $\psi$)
    $[\hat{\mathsf{P}}(\hat{\mathbf{y}}|\mathbf{x}),\ \nu_{\text{Nash}_1}] \leftarrow$ solveZeroSumGame$_{\hat{\mathsf{Y}}}(\mathbf{C})$
    $[\check{\mathbf{y}}_{\text{BR}},\ \check{\nu}_{\text{BR}}] \leftarrow$ findBestResponseStrategy($\hat{\mathsf{P}}(\hat{\mathbf{y}}|\mathbf{x})$, $\psi$)
    $\check{\mathsf{S}} \leftarrow \check{\mathsf{S}} \cup \check{\mathbf{y}}_{\mathsf{BR}}$
  until $(\nu_{\text{Nash}_1} = \check{\nu}_{\text{BR}})$
  return $[\hat{\mathsf{P}}(\hat{\mathbf{y}}|\mathbf{x}),\ \check{\mathsf{P}}(\check{\mathbf{y}}|\mathbf{x})]$
```

The size of the payoff matrix, $\mathbf{C}'$ from Equation 3.13, in the double oracle method is $\mathcal{O}(|\hat{\mathsf{S}}||\check{\mathsf{S}}|)$ as we depict in Table II. In the single oracle method linear equations of Equation 3.16 and Equation 3.17 corresponds to a matrix of size $\mathcal{O}(|\check{\mathsf{S}}|\mathsf{T}|\mathcal{Y}|)$. The complexity of efficient linear programs are about $\mathcal{O}(\mathsf{k}^{3.5})$ where k is the number of variables. Therefore, single oracle is efficient when the size of predictor's pure strategies in the double oracle is sufficiently large, then the added complexity of the linear program of the single oracle is compensated by the size reduction of the overall payoff matrix.

3.2.4 Solving Game in Terms of Pair-wise Marginal Probabilities

Finding the equilibrium in single oracle is still an iterative process with a search space for the adversary's mixed strategy of $|\mathcal{Y}|^{\mathsf{T}}$. Since the objective can be additively decomposed, as when finding the best response in the oracle methods as well as in the adversary objective in single oracle method, we can formulate the objective that further decouples the adversary's distribution from the full structure in Equation 3.17 to each node separately. For each edge connecting two adjacent nodes, we have a pairwise marginal probability $\check{P}(\check{y}_t, \check{y}_{t+1}|\mathbf{x_t}, \mathbf{x_{t+1}})$. The game value at each node $\mathbf{t}$ depends only on the pairwise marginals corresponding to the $\mathbf{t}$-th node, which are $\check{P}(\check{y}_t, \check{y}_{t+1}|\mathbf{x_t}, \mathbf{x_{t+1}})$ or $\check{P}(\check{y}_{t-1}, \check{y}_t|\mathbf{x_{t-1}}, \mathbf{x_t})$. The maximizer linear program then can be written in terms of, one of the two probabilities, $\check{P}(\check{y}_t, \check{y}_{t+1}|\mathbf{x_t}, \mathbf{x_{t+1}})$ with an additional constraint ensuring that marginal probabilities from the pairwise distributions are equal: $\sum_{\check{y}_{t+1}} \check{P}(\check{y}_t, \check{y}_{t+1}|\mathbf{x_t}, \mathbf{x_{t+1}}) = \sum_{\check{y}_{t-1}} \check{P}(\check{y}_{t-1}, \check{y}_t|\mathbf{x_{t-1}}, \mathbf{x_t})$.

$$\max_{\substack{\check{\mathbf{p}}_{12}, \check{\mathbf{p}}_{23}, \ldots, \check{\mathbf{p}}_{T-1,T}, \\ \nu_1, \nu_2, \ldots, \nu_T}} \quad \sum_{t=1}^{T} \nu_t \tag{3.18}$$

$$\begin{aligned}
\text{such that: } & \nu_t \le \mathbf{C}_{\hat{\mathbf{y}}_t,*} \check{\mathbf{p}}_{t-1,t} \quad \forall \hat{\mathbf{y}}_t \in \boldsymbol{\mathcal{Y}} \; \forall t \in \{2, ..., T\} \\
& \nu_1 \le \mathbf{C}_{\hat{\mathbf{y}}_1,*} \check{\mathbf{p}}_{1,2} \quad \forall \hat{\mathbf{y}}_1 \in \boldsymbol{\mathcal{Y}} \\
& \sum_{\check{y}_{t-1}} \check{\mathbf{p}}_{t-1,t} = \sum_{\check{y}_{t+1}} \check{\mathbf{p}}_{t,t+1} \quad \forall t \in \{2, ..., T-1\} \\
& \mathbf{1}^{\mathsf{T}} \check{\mathbf{p}}_{t-1,t} = 1. \quad \forall t \in \{2, ..., T\}
\end{aligned}$$

The number of variables is $\mathcal{O}(|\mathcal{Y}|^2\mathsf{T})$ since there are $|\mathcal{Y}|^2$ pairwise-marginals for each node. This is much less than the worst-case size in single oracle's $|\check{\mathsf{S}}| = |\mathcal{Y}|^\mathsf{T}$ and we do not require Algorithm 2 to iteratively search for the equilibrium. For a dataset with four classes and about sequences of length 31, Figure 8 shows the convergence speed of pairwise-marginal method compared to the single oracle method.

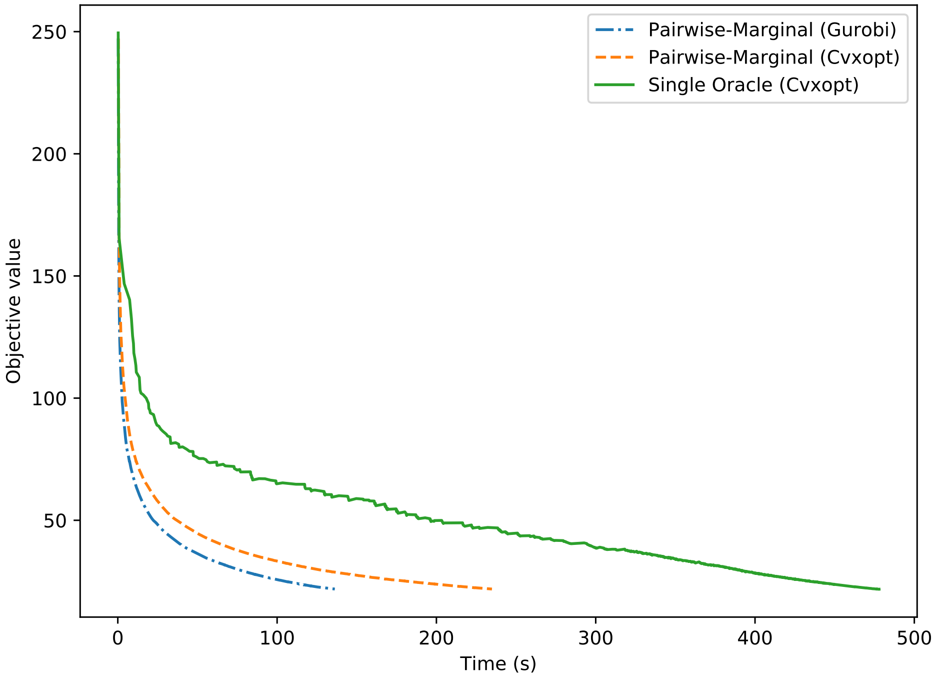


Figure 8: Comparison of convergence speed of Single Oracle and Pairwise-marginal method

We have used Gurobi (Gurobi Optimization, 2015) at first and later Cvxopt (Andersen et al., 2019) only to implement the single oracle in Python. For comparison, we show the convergence speed of pairwise-marginal implemented in Cvxopt as well, which shows that for the selected dataset, pairwise-marginal method is at least twice as fast. Also, noticeable is the time spent by the single oracle for iterative search of the equilibria by horizontal plateaus in the corresponding step-like plot.

Unfortunately, using pairwise choices for the adversary does not allow one to formulate predictor's linear program to obtain its mixed strategies. But during learning only the adversary's distributions (Equation 3.19) are needed. During prediction, however, if randomized assignment is not required then potential based assignment can be used, otherwise single oracle for a probability estimates is needed.

3.2.5 Learning via Convex Optimization

As shown in Theorem 3 in Section 3.1.3, the difference of the feature expectation under the adversary's distribution and the empirical feature expectation provides the gradient to optimize the Lagrangian parameters via convex optimization. The feature expectation of the sequence samples are computed using the following equation:

$$\begin{aligned}\mathbb{E}_{\check{\mathbf{P}}(\check{\mathbf{y}}|\mathbf{x})}\left[\Phi(\mathbf{x},\check{\mathbf{Y}})\right] &= \mathbb{E}_{\check{\mathbf{P}}(\check{\mathbf{y}}|\mathbf{x})}\left[\sum_{t=1}^{T-1}\phi(\mathbf{x},\check{y}_t,\check{y}_{t+1})\right] \\ &= \sum_{t=1}^{T-1}\sum_{y,y'}\check{P}(\check{Y}_t = y', \check{Y}_{t+1} = y|\mathbf{x},\theta)\phi(\mathbf{x},\check{y}_t,\check{y}_{t+1}).\end{aligned} \tag{3.19}$$

The empirical feature expectation is computed as $\tilde{c} = \theta\Phi(\mathbf{x}, \tilde{\mathbf{y}}) = \theta \sum_{t=1}^{T-1} \phi(\mathbf{x}, \tilde{y}_t, \tilde{y}_{t+1})$. Equation 3.19 only requires the adversary's distribution and therefore any of the above three methods can be used during learning. Algorithm 1 is then used to obtain the model parameters using stochastic gradient descent.

3.2.6 Consistency

Similar to adversarial cost-sensitive classification, adversarial sequence tagging (AST) also provides consistency guarantee.

Theorem 6. *Given that the sequence's probability distribution factors according to the chain independence assumptions:* $P(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{T} P(y_t|y_{t-1}, \mathbf{x}_{1:T})$, *and an arbitrarily rich feature representation,* $\psi(y_t, y_{t+1}, \mathbf{x}_{1:T})$, *the AST method provides the loss-optimal sequence tagging,* $\operatorname{argmin}_{\hat{\mathbf{y}}} \mathbb{E}_{P(\mathbf{Y}|\mathbf{x})} [loss(\hat{\mathbf{y}}, \mathbf{Y})]$.

Proof. The Lagrangian of Equation 3.12 gives, equivalently:

$$\min_{\psi(\cdot,\cdot)} \max_{\check{P}(\check{\mathbf{y}}|\mathbf{x})} \min_{\hat{P}(\hat{\mathbf{y}}|\mathbf{x})} \mathbb{E}_{P(\mathbf{x},\mathbf{y})} \Big[\mathbb{E}_{\hat{P}(\hat{\mathbf{y}}|\mathbf{x})\check{P}(\check{\mathbf{y}}|\mathbf{x})} \Big[\text{loss}(\hat{\mathbf{Y}}, \check{\mathbf{Y}}) \\ + \psi(\mathbf{X}, \check{\mathbf{Y}}) - \psi(\mathbf{X}, \mathbf{Y}) \Big| \mathbf{X} \Big] \Big] \tag{3.20}$$

$$
\begin{aligned}
&\overset{(a)}{=} \max_{\check{\mathsf{P}}(\check{\mathbf{y}}|\mathbf{x})} \min_{\psi(\cdot,\cdot)} \Bigg(\mathbb{E}_{\mathsf{P}(\mathbf{x},\mathbf{y})\check{\mathsf{P}}(\check{\mathbf{y}}|\mathbf{x})} \Big[\psi(\mathbf{X},\check{\mathbf{Y}}) - \psi(\mathbf{X},\mathbf{Y})\Big] \\
&\qquad\qquad + \min_{\hat{\mathsf{P}}(\hat{\mathbf{y}}|\mathbf{x})} \mathbb{E}_{\mathsf{P}(\mathbf{x})\check{\mathsf{P}}(\check{\mathbf{y}}|\mathbf{x})\hat{\mathsf{P}}(\hat{\mathbf{y}}|\mathbf{x})} \Big[\mathrm{loss}(\hat{\mathbf{Y}},\check{\mathbf{Y}})\Big]\Bigg) \\
&\overset{(b)}{=} \min_{\hat{\mathsf{P}}(\hat{\mathbf{y}}|\mathbf{x})} \mathbb{E}_{\mathsf{P}(\mathbf{x},\mathbf{y})\hat{\mathsf{P}}(\hat{\mathbf{y}}|\mathbf{x})} \Big[\mathrm{loss}(\hat{\mathbf{Y}},\mathbf{Y})\Big] \\
&\overset{(c)}{=} \mathbb{E}_{\mathsf{P}(\mathbf{x})}\Big[\min_{\hat{\mathbf{y}}} \mathbb{E}_{\mathsf{P}(\mathbf{y}|\mathbf{x})}\Big[\mathrm{loss}(\hat{\mathbf{y}},\mathbf{Y})\Big|\mathbf{X}\Big]\Big],
\end{aligned} \tag{3.21}
$$

the transformation steps are:

(a) Lagrangian duality allows to swap min and max. The expectation terms are rearranged, since $\hat{\mathsf{P}}(\hat{\mathbf{y}}|\mathbf{x})$ terms are not in the potential terms, it can be moved to the end.

(b) If $\check{\mathsf{P}}(\mathbf{y}|\mathbf{x}) \neq \mathsf{P}(\mathbf{y}|\mathbf{x})$, $\min_{\psi}$ can make the value of Equation 3.21 unboundedly negative. Therefore adversary distribution must equal to the true distribution and thus the potential difference becomes zero.

(c) From probabilistic to non-probabilistic decision.

This is, by definition, the set of risk-minimizing predictions. □

Thus, given any true distribution of sequence data, $\mathsf{P}(\mathbf{y},\mathbf{x})$, a consistent predictor minimizing the sequence loss will be obtained when the feature representation is rich enough to sufficiently capture the sequence relationships.

CHAPTER 4

APPLICATIONS

(Results in section 4.1 were published in Asif, Kaiser, Wei Xing, Sima Behpour, and Brian D. Ziebart. "Adversarial Cost-Sensitive Classification." In *UAI*, pp. 92-101. 2015. (Asif et al., 2015), and FAQ result from section 4.2 in Jia Li, Kaiser Asif, Hong Wang, Brian D. Ziebart, and Tanya Y. Berger-Wolf. "Adversarial Sequence Tagging." In *IJCAI*, pp. 1690-1696. 2016. (Li* et al., 2016))

4.1 Cost-sensitive Classification

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

4.1.1 Datasets

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

4.1.2 Methodology

Costs for the classification task are not predefined, so random cost matrices are used to compare performance of the algorithms. We conduct 10 cost-sensitive classification tasks for

TABLE III: Evaluation datasets and dataset characteristics.

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

each dataset to get an average performance. We generate confusion cost matrices, **C**, for each task by:

1. Assigning all correct classifications a cost of zero ($C_{i,i} = 0, \forall i$); and
2. Sampling the remaining elements of the cost matrix from the uniform distribution ($C_{i,j} \sim U[0, 1], \forall i \neq j$).

For each classification task, we split the data into training and testing sets as described in Table III. We measure the expected cost of each method averaged over each of the 10 tasks.

4.1.3 Comparison Methods

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

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

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

- **Cost-Sensitive One-Versus-One (CSOVO)**: We employ the LIBSVM (Chang and Lin, 2011) implementation of the CSOVO SVM approach described in section 2.2. Our experiments use quadratic kernels (Chang and Lin, 2011), $K(u, v) = (\gamma_1 u'v + \gamma_0)^2$ to match the expressiveness of our approach. We run five-fold cross validation on the training set of every dataset to choose quadratic kernel parameters (shown in Table Table IV), and then we use these best parameters to train from the training set and construct the final classifier model. We use the default tolerance of termination criterion, 0.001, for most of the datasets except *image segmentation* and *shuttle*, which required a less sensitive criterion to converge. Finally, we evaluate the CSOVO performance by measuring the prediction cost on the test data.

- **Cost-Sensitive One-Versus-All (CSOVA)**: We similarly employ the LIBSVM implementation of the CSOVA SVM approach described in section 2.1. Our methodology matches that of CSOVO for cross-validation (parameters shown in Table Table IV), training, and testing.
- **Structured SVM** (SVM-Struct): We employ the Large Scale Structured SVM (SVM LS) software package (Branson et al., 2013) to obtain a multiclass cost-sensitive predictor based on the additive cost-sensitive hinge loss of Equation 2.7. SVM LS applies online subgradient methods (Ratliff et al., 2007) and sequential order optimization (Shalev-Shwartz et al., 2011) to improve efficiency. We evaluate the Online Dual Ascent (ODA) algorithm (Branson et al., 2013) as well as the Stochastic Gradient Descent (SGD) method for the purpose of our cost-sensitive experiments. We employ a trade-off parameter α of 100.

4.1.4 Results

Figure 9 shows the average loss incurred by each approach on the 13 different datasets. Our method generally performs well on all of the datasets except *wine* and *libras* datasets and has a similar performance with boosting. SVM methods except SVM-CSOVO are strong on some of

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

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

the datasets (*optdigits*, *pendigits*, *wine* and *libras*). For many datasets, the performance of the reduction-based SVM approaches is significantly worse than our approach and boosting and the multi-class structured SVM approach. The multi-class structured SVM approach specifically is significantly worse than our method on many of the datasets (*satimage*, *shuttle*, *vehicle*, *breast tissue*, *pendigits*, and *vertebral*), while only significantly better on the *optdigits* dataset.

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

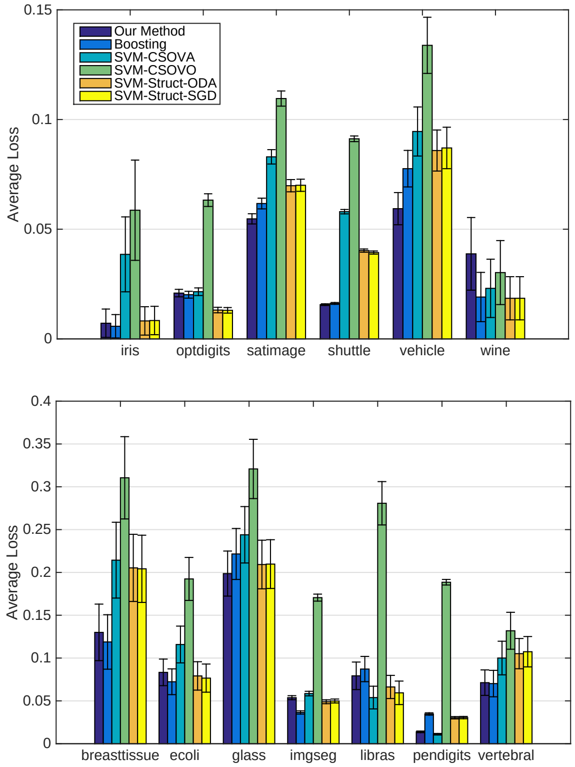


Figure 9: The average loss of predictions for the datasets of Table Table III.

boosting is significantly better, while our method is significantly better for the remaining four (*satimage*, *shuttle*, *vehicle*, and *pendigits*).

We compare the average loss of the prediction methods aggregated over all of the datasets in Figure Figure 10, showing that on average our method provides lower cost predictions. It is important to note that as an ensemble method, boosting is able to implicitly consider a much richer feature space than our approach. For classification, SVMs are often only comparable when incorporating kernels that can also implicitly consider richer feature spaces. Thus, exceeding the performance of the state-of-the-art boosting method using only quadratic features is a

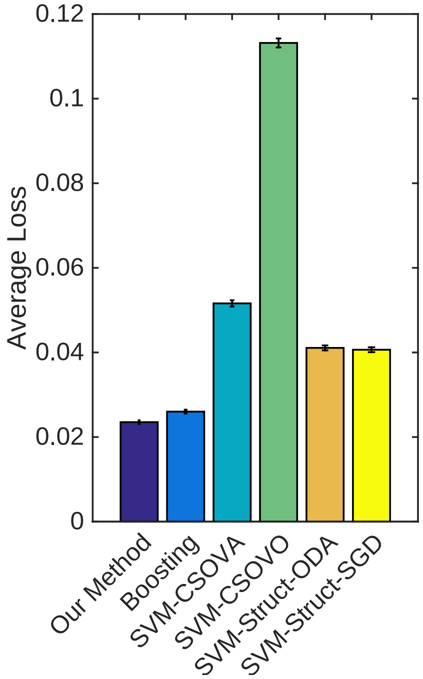


Figure 10: Average loss of predictions across all datasets of Table Table III.

significant demonstration of our method. The comparisons with the structured SVM method, which considers an identical feature space, illustrates the general benefit our approach provides by adversarially approximating the training data rather than convexly approximating the loss function.

4.2 Sequence Tagging

As an example of the experiments done with adversarial sequence tagging (AST), we describe here the Faq Segmentation task from our publication on Adversarial Sequence Tagging (Li* et al., 2016). The Faq Segmentation dataset (McCallum et al., 2000) contains 48 Frequently Asked Questions (FAQs) downloaded from the Internet. 26 are used for training and 22 for testing.

Each line in the document is labeled with four possible labels: head, question, answer, and tail. 24 Boolean features are generated for each line.

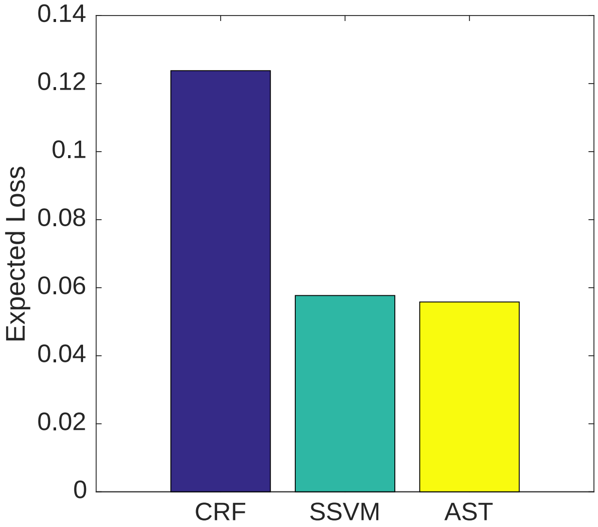


Figure 11: Expected loss AST compared with CRF and SVM

We compare our method against linear chain CRF (Sarawagi and Cohen, 2004) and Structural SVM (SSVM). The features for CRF and AST (our method) are based on transition between sequence variables $\phi(y_t, y_{t+1})$ and data input at the variables $\phi(x_t, y_t)$. For SSVM we use SVM^{hmm} from the SVM^{light} package (Joachims, 1999). We include first order tag sequence as features here. For AST we use double oracle for training and testing. We start with pure strategies of the same label for the whole sequence $\{11...1, 22...2, ...\}$. To solve the linear programs we use Gurobi (Gurobi Optimization, 2015).

We use 10% of the data for cross-validation to select parameters: regularization for CRF and trade-off weight C for SSVM.

The result shows that our method outperforms both CRF (average loss 0.124) and SSVM (0.058) by having an average loss of 0.0558.

4.2.1 Adversarial Sequence Tagging in Intelligent Welding

(This work has been submitted to 9th International Conference on Acoustic Emission as (Asif et al., 2019))

As a direct application of sequence tagging, we apply the pairwise joint probability based algorithm to an intelligent welding system. In gas welding two or more metal parts are joined together using filler material.

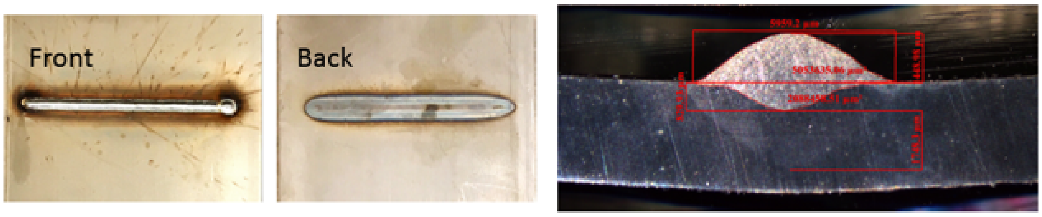


Figure 12: Weld on a metal plate. From the left: front, back, and cross-section

Figure 12 shows ideal weld quality and Figure 13 shows a robotic welding-arm used in automated welding systems. Due to various parameters of the welding system, weld qualities may vary. As the welding is a continuous process, the quality detection problem can be considered

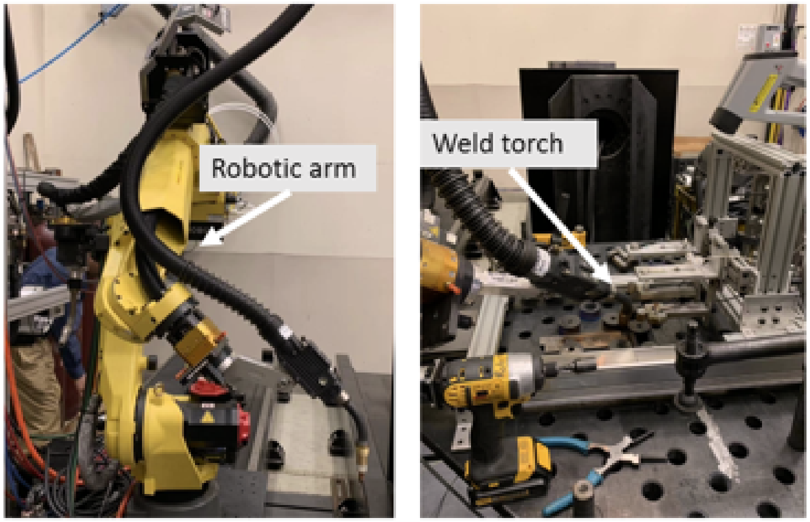


Figure 13: Weld robot

as a sequence tagging problem. The results shown in this work compare the sequence tagging method with simpler classification method, e.g. logistic regression. We collected the data and therefore describe here the detailed process of feature generation and experiment evaluation.

The goal of welding is to join two or more metal parts using a weld material. A defect where excessive weld material penetrates the metal completely is called burn-through (Figure 14). Another weld defect is porosity where impurity in the gas flow causes bubbles trapped into the weld (Figure 15).

The quality of a weld is usually determined by human inspection and often as a post-processing, mostly involving destructive measures like cutting the weld. All these methods are obstacle to an automated welding system since the decision cannot be made run-time and independent of human observation. Therefore an automated weld-quality detection system

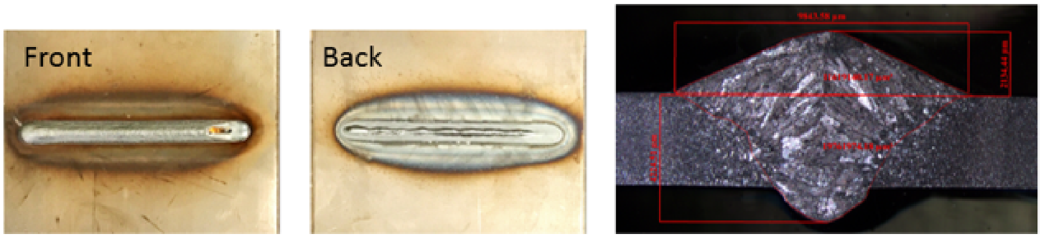


Figure 14: Burn-through weld on a metal plate. From the left: front, back, and cross-section

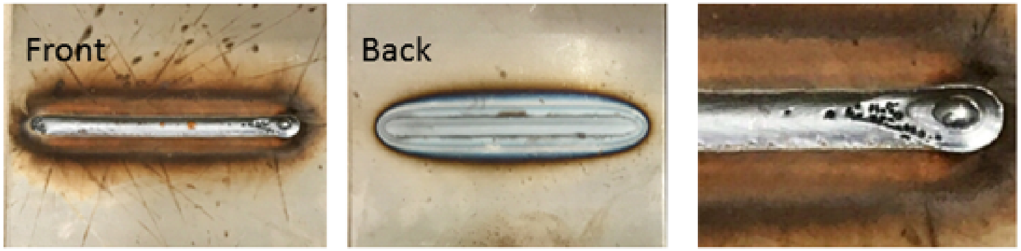


Figure 15: Porosity weld on a metal plate. From the left: front, back, and close-up view of porosity bubbles

using acoustic emission is sought where acoustic emission is captured during welding using various sensors and along with welding input parameters used to predict the quality of the weld. Machine learning has previously been employed in welding. For example, using input parameters performance of welding trainees have been evaluated (Kumar et al., 2018). Quality of weld has not been addressed. Relationship of the acoustic emission and weld-parameters with the weld-defects has been established in post-processing, but machine learning had not been employed for real-time monitoring (Zhang et al., 2018). Neural networks and support vector machines have been used to predict weld quality based on acoustic emission (Sumesh et al., 2015b; Sumesh et al., 2015a). Our method varies in two ways: the number of target classes simultaneously used is higher and by considering the weld to be a continuous process instead of discrete data-points in time, i.e., we consider previously predicted data-points' weld quality while predicting current data-point. Thus the prediction will be more consistent to the weld area if any spurious noise in the feature occurs.

4.2.1.1 Data Description

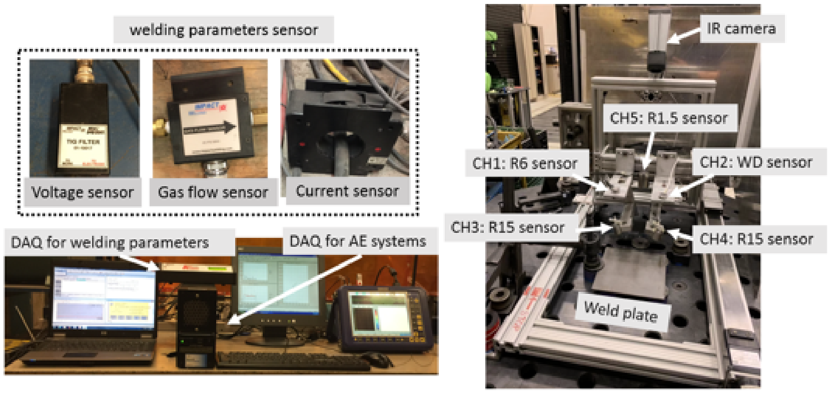


Figure 16: Weld data collection sensors.

During the welding process, input parameters such as amperage, voltage, speed, and gas flow rate are recorded. From these values, heat-input is computed. While heat-input is already highly correlated to penetration and burn-through, acoustic emission (AE) is used as an additional attribute to infer weld quality.

TABLE V: Dataset description of the welding experiment. Number of samples in each group are shown in parentheses.

Sample Group	**Wire Speed (in/min)**	**Gas Flow (ft^3/h)**	**Travel Speed (mm/s)**	**Heat Input (KJ/mm)**	**Expected Quality**
JD-P1-TS (3)	200	40	11	0.38	Good Weld
JD-P2-TS (3)	160	40	11	0.30	Good Weld
JD-P3-TS (3)	120	40	11	0.20	Good Weld
JD-P4-TS (3)	100	40	11	0.18	Good Weld
JD-P5-TS (3)	240	40	11	0.49	Penetration
JD-P6-TS (3)	260	40	11	0.52	Penetration
JD-P7-TS (3)	280	40	11	0.58	Onset of Burn-through
JD-P8-TS (3)	300	40	11	0.60	Burn-through
JD-P1-PO1 (3)	200	25	11	0.39	Good Weld
JD-P1-PO2 (3)	200	21	11	0.40	Porosity
JD-P1-PO3 (5)	200	12	11	0.39	Porosity
JD-P7-PO3 (5)	280	12	11	0.56	Porosity + Burn-through
JD-P7-PO4 (1)	280	56	11	0.56	Porosity + Burn-through
JD-P7-PO5 (1)	280	59	11	0.56	Porosity + Burn-through

At first, in order to generate different weld-qualities, weld-samples with various configurations were generated using a gas tungsten arc welding system (GTAW). We changed amperage, voltage and speed to change heat-input that in turn created different penetration levels. We then used clustering methods (K-means and Hierarchical) to find different groups. However, correct configurations to generate distinctive clusters were not found. Perhaps largely due to the fact that GTAW is a quieter process and thus AE signals are not significant enough. Afterwards, we collected gas metal arch weld data (GMAW) in an industrial setting. There AE sound change between different quality welds was noticeable even by human ear. Therefore, we proceeded with GMAW welding system for our experiment.

Table V shows the configurations how different quality welds have been generated in GMAW. The target categories selected for modeling are “Good Weld”, “Penetration”, “Burn-through”, and “Porosity”. Input parameters are recorded with 0.1 second intervals. Acoustic emission is captured in two sets of data: Absolute energy and Average Signal Level (ASL) are recorded in 0.02 second intervals, these are called time-driven data (TDD). And, there are hit-driven data (HDD) which captures values over specific thresholds without fixed time-intervals Figure 17, the features we use are counts, amplitude, frequency centroid, and peak frequency.

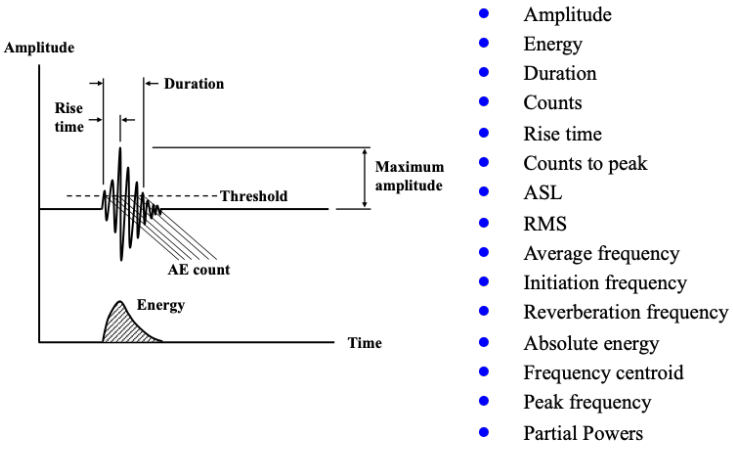


Figure 17: Hit-Driven Data (HDD) features.

We first synchronize the TDD and interpolated input parameters as 0.02 second intervals. We merge the HDD features in feature generation phase.

4.2.1.2 Feature Generation

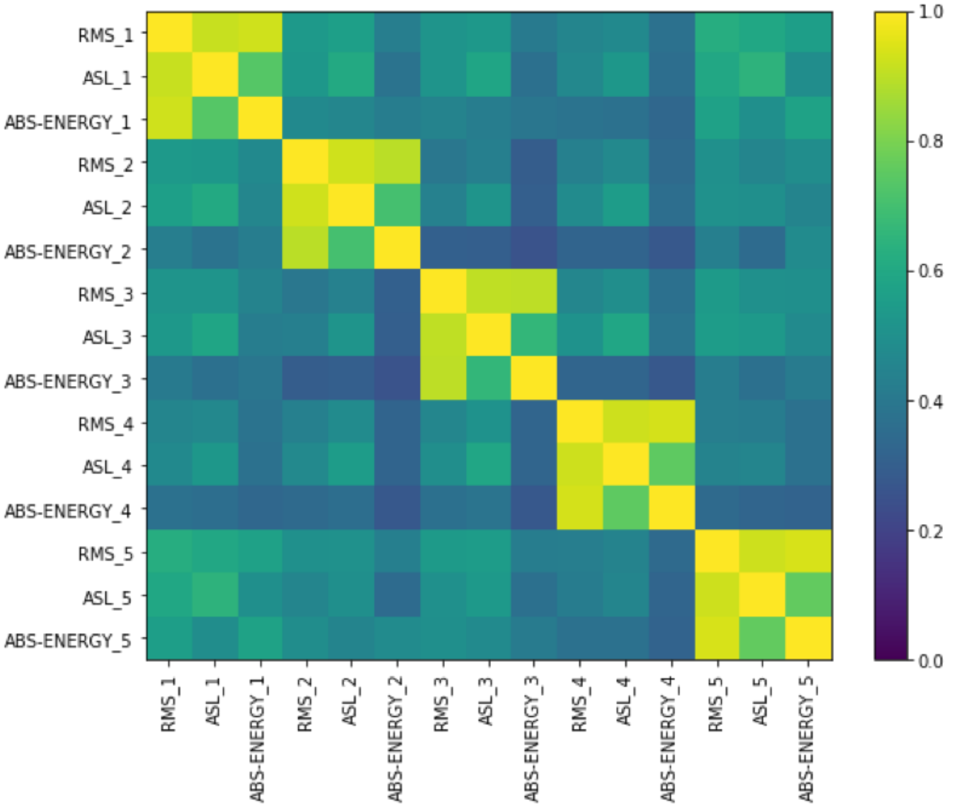


Figure 18: Correlations of time-driven (TDD) features. Darker means lower correlation. Channels 2 and 3 have lower correlations compared to other pairs.

After synchronizing, we first select two acoustic emission channels that have lower correlations (Figure 18). We select two channels to keep number of features low to avoid overfitting. With extracted features from two channels and quadratic transformation, we eventually have above 900 features. Also noticeable from Figure 18 is that RMS is highly correlated to corresponding channel's ASL and absolute energy, therefore, we exclude this feature from the channels.

Afterwards, we apply signal smoothing with a rolling window size 10, and then group each 10 time-steps together. This gives us about 24 to 31 time-steps per sample. From each group-window, the minimum, the maximum and the average values are selected. In addition, *rate of accumulation of energy* is computed as $\sum_t (tx_t - t\mu_x)$, which gives the area in Figure 19.

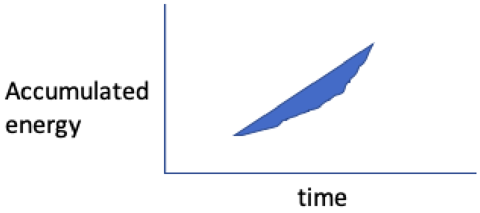


Figure 19: Rate of accumulation of energy

HDD features are taken from the selected features and merged in the 0.2 second window as averages or 0 if absent. These features, mentioned here and in data description, have been selected via histogram analysis from a larger pool of features. Once generated, we use min-max scaling,$(x_i - min_i)/(max_i - min_i)$, to make the feature values between 0 and 1, and then do a quadratic transformation to capture any existing relations between the features.

4.2.1.3 Results

TABLE VI: Results of the welding experiment. Bold percentages indicate the expected weld quality from Table V: Good Weld (G), Penetration (Pn), Burn-through (B), Porosity (Pr). JD-P7 samples were not seen in the training data and does not have distinctive labels.

	Adversarial Sequence Tagging				**Logistic Regression**			
Sample	**G**	**Pn**	**B**	**Pr**	**G**	**Pn**	**B**	**Pr**
JD-P1-TS-3	**100**	0	0	0	**96.8**	0	0	3.23
JD-P5-TS-3	3.23	**96.77**	0	0	0	**100**	0	0
JD-P7-TS-1	0	0	100	0	0	9.68	90.32	0
JD-P7-TS-2	3.23	0	96.77	0	0	3.23	96.77	0
JD-P1-PO1-1	**83.9**	16.13	0	0	**83.9**	0	0	16.13
JD-P1-PO1-2	**96.8**	0	3.23	0	**93.6**	0	0	6.45
JD-P1-PO1-3	**77.4**	12.9	0	9.68	**71**	0	0	29.03
JD-P7-PO3-1	0	0	96.77	3.23	0	0	19.35	80.65
JD-P7-PO3-2	0	0	96.77	3.23	0	0	0	100
JD-P7-PO3-3	0	0	48.39	51.61	0	0	16.13	83.87
JD-P7-PO3-4	0	0	90.32	9.68	0	0	9.68	90.32
JD-P7-PO3-5	0	0	93.55	6.45	0	0	0	100
JD-P7-PO4	0	0	87.1	12.9	0	0	35.48	64.52
JD-P7-PO5	0	0	38.71	61.29	0	0	67.74	32.26

We compare our adversarial sequence tagging (AST) method with multinomial logistic regression (LR). For each sample, from the 24 to 31 congregated points, we look at the percentage of points classified as each to the classes. For logistic regression we consider each data point as discrete sample, whereas for AST full sequence represent one sample. For space limitation we exclude from Table VI the samples that have full correct classifications.

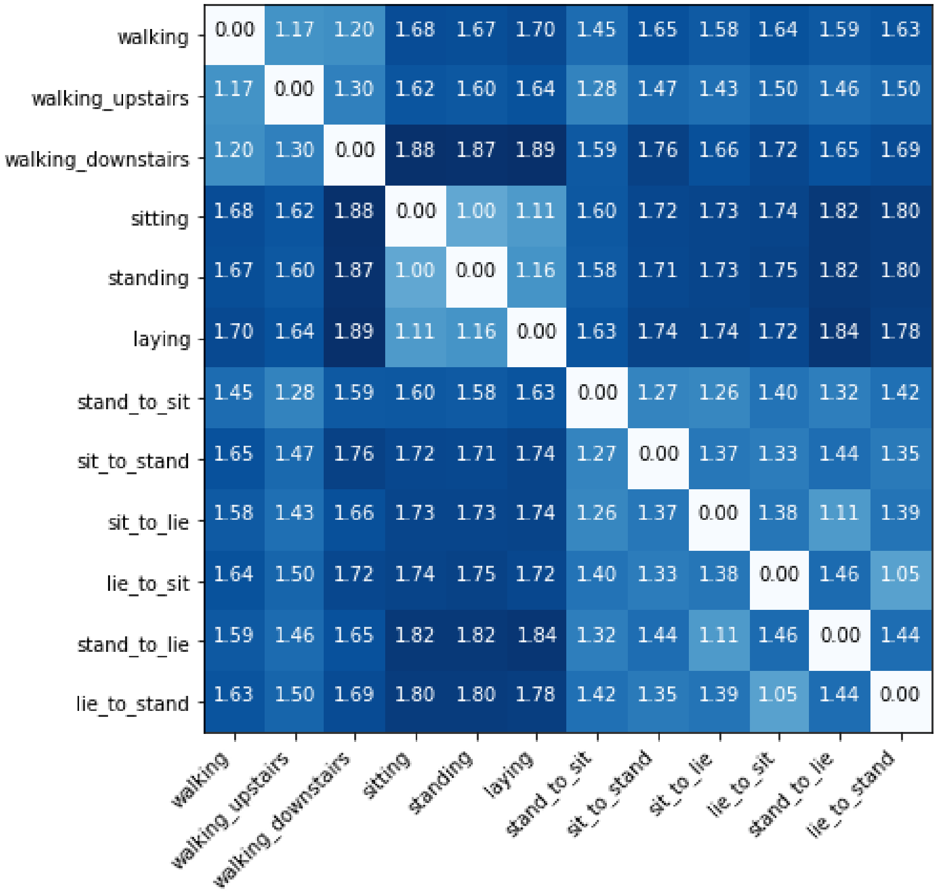
We consider 5% leniency per sample. and consider either of Penetration and Burn-through as correct category for samples JD-P7-TS (those samples had heavy melting and could fall between penetration and burn-through classes), and for JD-P7-PO samples let Burn-through or Porosity be correct since both quality were seen in inspection. Then AST fails to correctly label samples JD-P1-PO1-1 and JD-P1-PO1-3, whereas LR additionally fails at JD-P1-PO1-2. Therefore the macro-accuracy in terms of correctly labeled whole samples are 94.12% and 91.18% respectively.

4.2.2 Adversarial Cost-sensitive Sequence Tagging

To demonstrate a cost-sensitive sequence tagging application, we use dataset Smartphone-Based Recognition of Human Activities and Postural Transitions (Reyes Ortiz et al., 2015) from UCI repository. The data contains activity data collected via smartphone sensors from 30 subjects with 12 labeled classes in 561 extracted features. The dataset provides train-test split, but are not in sequence representation. We separate the data first by subject id and then randomly split between 20 and 200 lengths.

To generate a cost-matrix, we compute distances (scaled) of the classes based on Euclidian distance (Figure 20). As expected stationary activity to walking has higher cost than others.

We compare our method with structured-SVM (Finley and Joachims, 2007), we use the misclassification cost $\Delta(\mathbf{y}, \mathbf{y}_i) = \sum_{t=1}^{T} C_{y_t, y_{i,t}}$ in Equation 2.23. For both structured-SVM and Adversarial Sequence Tagging, we train a model on the cost-matrix and another 0-1 loss



	walking	walking_upstairs	walking_downstairs	sitting	standing	laying	stand_to_sit	sit_to_stand	sit_to_lie	lie_to_sit	stand_to_lie	lie_to_stand
walking	0.00	1.17	1.20	1.68	1.67	1.70	1.45	1.65	1.58	1.64	1.59	1.63
walking_upstairs	1.17	0.00	1.30	1.62	1.60	1.64	1.28	1.47	1.43	1.50	1.46	1.50
walking_downstairs	1.20	1.30	0.00	1.88	1.87	1.89	1.59	1.76	1.66	1.72	1.65	1.69
sitting	1.68	1.62	1.88	0.00	1.00	1.11	1.60	1.72	1.73	1.74	1.82	1.80
standing	1.67	1.60	1.87	1.00	0.00	1.16	1.58	1.71	1.73	1.75	1.82	1.80
laying	1.70	1.64	1.89	1.11	1.16	0.00	1.63	1.74	1.74	1.72	1.84	1.78
stand_to_sit	1.45	1.28	1.59	1.60	1.58	1.63	0.00	1.27	1.26	1.40	1.32	1.42
sit_to_stand	1.65	1.47	1.76	1.72	1.71	1.74	1.27	0.00	1.37	1.33	1.44	1.35
sit_to_lie	1.58	1.43	1.66	1.73	1.73	1.74	1.26	1.37	0.00	1.38	1.11	1.39
lie_to_sit	1.64	1.50	1.72	1.74	1.75	1.72	1.40	1.33	1.38	0.00	1.46	1.05
stand_to_lie	1.59	1.46	1.65	1.82	1.82	1.84	1.32	1.44	1.11	1.46	0.00	1.44
lie_to_stand	1.63	1.50	1.69	1.80	1.80	1.78	1.42	1.35	1.39	1.05	1.44	0.00

Figure 20: Cost-matrix for Human Activity Recognition

(Hamming loss), and then evaluate on the cost-matrix to show that models trained with the cost-matrix have lower total incurred costs (Table VII).

For comparison with non-sequence prediction, Logistic Regression model is used. For the 0-1 loss model, the output of Logistic Regression is taken and total cost is computed using the cost-matrix. For the cost-sensitive model, the probability of the Logistic Regression model is used to obtain the Bayes optimal prediction for the cost-sensitive prediction.

TABLE VII: Total misclassification cost incurred using cost from Figure 20 evaluated with models trained using the cost and using 0-1 (Hamming) loss

	Trained with 0-1 loss	**Trained with cost**
Structured SVM	136.28	131.31
Adversarial Sequence Tagging	106.99	102.15
Logistic Regression	(non-Bayes) 199.32	(Bayes) 198.85

We can see that AST outperforms other methods in both cases. Also, for all methods, the cost-sensitive model minimizes the total incurred cost better than the corresponding model trained without taking costs into account.

CHAPTER 5

CONCLUSION AND FUTURE WORK

In classification problems, depending on the nature of the data or the target problem, solely optimizing accuracy may not always be desired. If misclassifying one class has a higher real world loss compared to other classes, we may want to assign a higher cost to that class and perform cost-sensitive classification instead of maximizing accuracy. With this goal, we have developed a general adversarial framework to perform classification. We incorporated cost-sensitivity in this framework. In Section 3.1 we showed the detailed development of the adversarial method and showed its mathematical properties. We described how we can formulate a classification task as a convex optimization problem and how to solve this optimization. We showed that our method is Fisher consistent while multiclass SVM is not. Using this framework, we have performed cost-sensitive classification using randomly generated costs with publicly available UCI classification datasets. We compared our method with state-of-the-art methods (Section 4.1) and showed that our method was competitive in some cases and outperformed others. In Section 3.2 we extended our adversarial approach to sequence tagging problem. We discussed different approaches to solve a linear sequence tagging problem with the adversarial framework. We showed the general case of 0-1 loss (accuracy) in sequence tagging in Section 4.2 and then a cost-sensitive sequence tagging result in Section 4.2.2.

In the future, I want to explore other areas, for example, cost-sensitivity in deep neural networks. Early cost-sensitive deep networks have been studied for the class imbalance problem

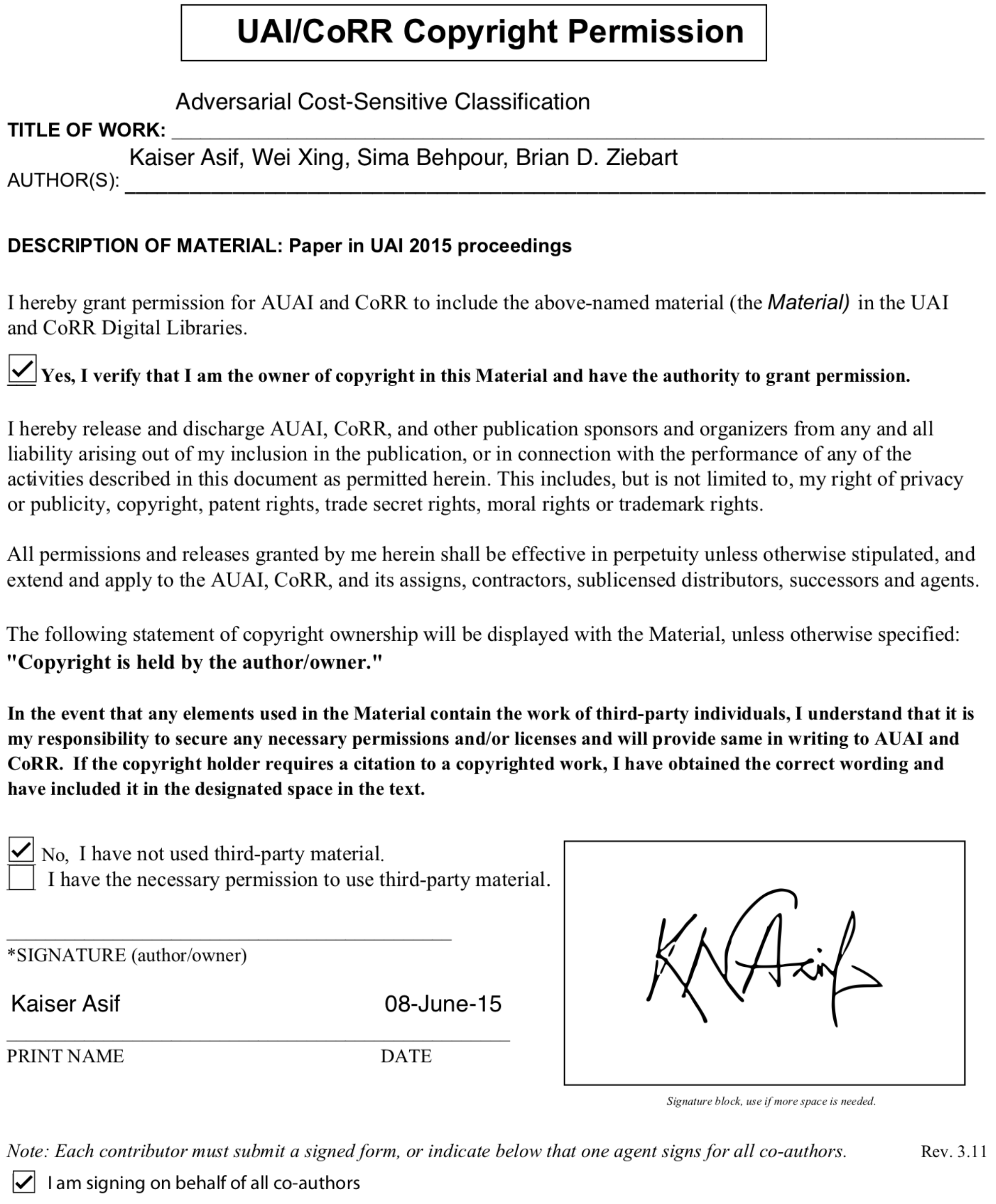
(Zhou and Liu, 2006) using resampling and threshold based classification. For general cost-matrices (Chung et al., 2015) follows one-sided SVM method by (Tu and Lin, 2010) but uses the smooth-SVM loss to replace the max function to log-exponential for differentiability. The limitation of this method may be that the loss does not contain the direct cost-matrix. (Khan et al., 2018) studies several cost-sensitive losses including mean squared error, SVM and softmax where the output of last layer is weighted by the classification cost of the corresponding training sample. (Tang, 2013) have shown that using hinge loss (SVM) in place of softmax gives better results. As our model is similar to SVM, we can follow these approaches and replace the SVM layer with our cost-sensitive adversarial loss.

Another direction can be in exploring ways to learn the costs for cost-sensitive classification. Currently we use arbitrary cost-matrices to evaluate the performance of the classifiers, but we should rather learn a cost-matrix that aligns with the task at hand. One method could be first compute cost via distances of class clusters and then show experts misclassified samples and update the cost-matrix iteratively based on feedback. Another interesting are is early event-detection where a event happens over a period of time and the task is to identify as early as possible. For example, human activity needs to be anticipated by robots to have better interaction, an earlier detection enables smoother interaction (Koppula and Saxena, 2016; Hoai and De la Torre, 2014), in a sequence of medical diagnosis the earlier a disease is detected the better (Xing et al., 2008). (Hoai and De la Torre, 2014) use maximum margin structured prediction formulation for early event detection. Therefore, our sequence tagging method can easily be applied in the area. We can also follow the abstention based prediction using adversarial method (Fathony et al., 2018) along with sequence tagging method.

APPENDIX

APPENDIX A

Permission to reuse publications from UAI and IJCAI.



UAI/CoRR Copyright Permission

TITLE OF WORK: Adversarial Cost-Sensitive Classification

AUTHOR(S): Kaiser Asif, Wei Xing, Sima Behpour, Brian D. Ziebart

DESCRIPTION OF MATERIAL: Paper in UAI 2015 proceedings

I hereby grant permission for AUAI and CoRR to include the above-named material (the *Material)* in the UAI and CoRR Digital Libraries.

☑ **Yes, I verify that I am the owner of copyright in this Material and have the authority to grant permission.**

I hereby release and discharge AUAI, CoRR, and other publication sponsors and organizers from any and all liability arising out of my inclusion in the publication, or in connection with the performance of any of the activities described in this document as permitted herein. This includes, but is not limited to, my right of privacy or publicity, copyright, patent rights, trade secret rights, moral rights or trademark rights.

All permissions and releases granted by me herein shall be effective in perpetuity unless otherwise stipulated, and extend and apply to the AUAI, CoRR, and its assigns, contractors, sublicensed distributors, successors and agents.

The following statement of copyright ownership will be displayed with the Material, unless otherwise specified:
"Copyright is held by the author/owner."

In the event that any elements used in the Material contain the work of third-party individuals, I understand that it is my responsibility to secure any necessary permissions and/or licenses and will provide same in writing to AUAI and CoRR. If the copyright holder requires a citation to a copyrighted work, I have obtained the correct wording and have included it in the designated space in the text.

☑ No, I have not used third-party material.
□ I have the necessary permission to use third-party material.

*SIGNATURE (author/owner)

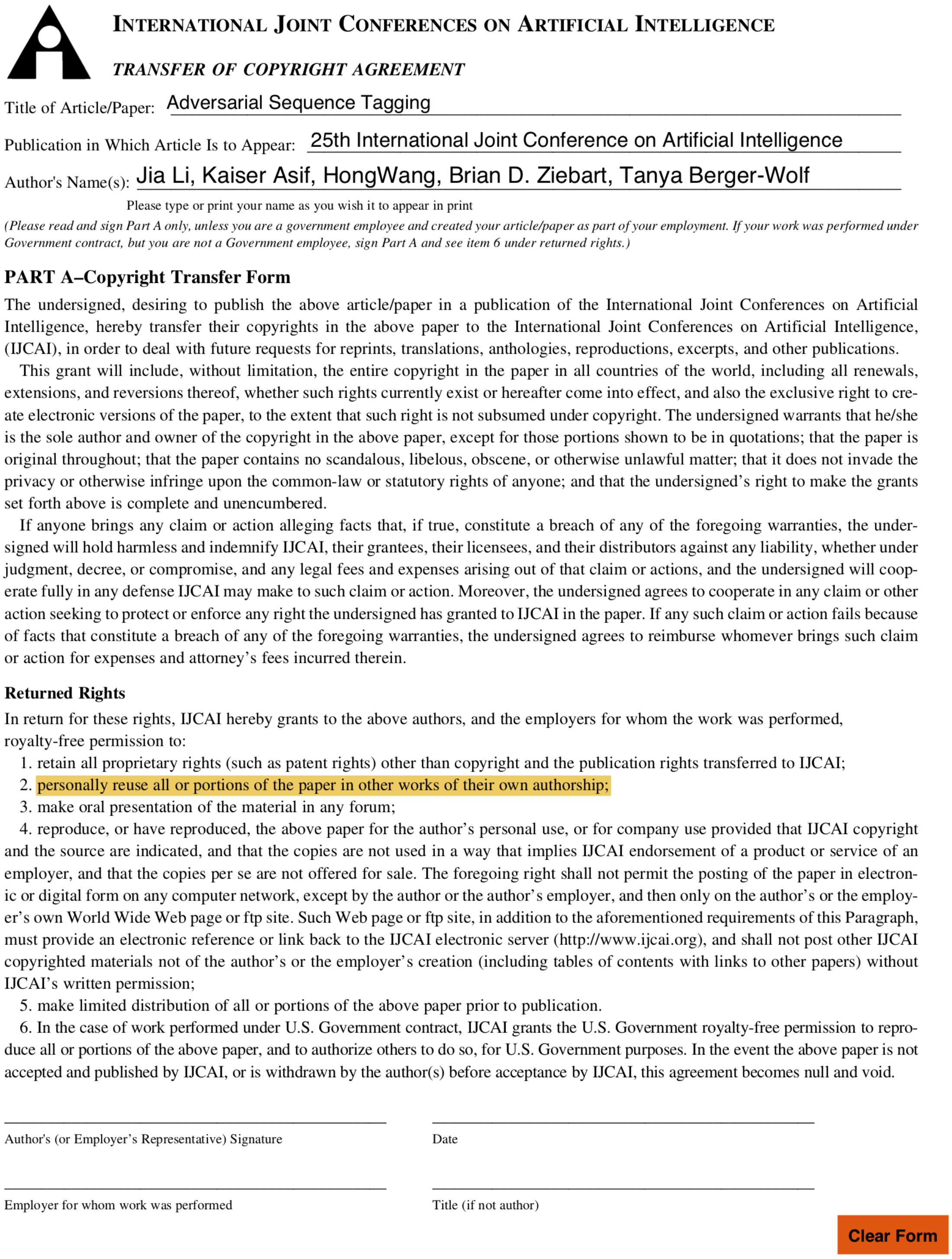
Kaiser Asif — 08-June-15
PRINT NAME — DATE

Signature block, use if more space is needed.

Note: Each contributor must submit a signed form, or indicate below that one agent signs for all co-authors. Rev. 3.11

☑ I am signing on behalf of all co-authors

APPENDIX (Continued)



INTERNATIONAL JOINT CONFERENCES ON ARTIFICIAL INTELLIGENCE

TRANSFER OF COPYRIGHT AGREEMENT

Title of Article/Paper: Adversarial Sequence Tagging

Publication in Which Article Is to Appear: 25th International Joint Conference on Artificial Intelligence

Author's Name(s): Jia Li, Kaiser Asif, HongWang, Brian D. Ziebart, Tanya Berger-Wolf

Please type or print your name as you wish it to appear in print

(Please read and sign Part A only, unless you are a government employee and created your article/paper as part of your employment. If your work was performed under Government contract, but you are not a Government employee, sign Part A and see item 6 under returned rights.)

PART A–Copyright Transfer Form

The undersigned, desiring to publish the above article/paper in a publication of the International Joint Conferences on Artificial Intelligence, hereby transfer their copyrights in the above paper to the International Joint Conferences on Artificial Intelligence, (IJCAI), in order to deal with future requests for reprints, translations, anthologies, reproductions, excerpts, and other publications.

This grant will include, without limitation, the entire copyright in the paper in all countries of the world, including all renewals, extensions, and reversions thereof, whether such rights currently exist or hereafter come into effect, and also the exclusive right to create electronic versions of the paper, to the extent that such right is not subsumed under copyright. The undersigned warrants that he/she is the sole author and owner of the copyright in the above paper, except for those portions shown to be in quotations; that the paper is original throughout; that the paper contains no scandalous, libelous, obscene, or otherwise unlawful matter; that it does not invade the privacy or otherwise infringe upon the common-law or statutory rights of anyone; and that the undersigned's right to make the grants set forth above is complete and unencumbered.

If anyone brings any claim or action alleging facts that, if true, constitute a breach of any of the foregoing warranties, the undersigned will hold harmless and indemnify IJCAI, their grantees, their licensees, and their distributors against any liability, whether under judgment, decree, or compromise, and any legal fees and expenses arising out of that claim or actions, and the undersigned will cooperate fully in any defense IJCAI may make to such claim or action. Moreover, the undersigned agrees to cooperate in any claim or other action seeking to protect or enforce any right the undersigned has granted to IJCAI in the paper. If any such claim or action fails because of facts that constitute a breach of any of the foregoing warranties, the undersigned agrees to reimburse whomever brings such claim or action for expenses and attorney's fees incurred therein.

Returned Rights

In return for these rights, IJCAI hereby grants to the above authors, and the employers for whom the work was performed, royalty-free permission to:

1. retain all proprietary rights (such as patent rights) other than copyright and the publication rights transferred to IJCAI;
2. personally reuse all or portions of the paper in other works of their own authorship;
3. make oral presentation of the material in any forum;
4. reproduce, or have reproduced, the above paper for the author's personal use, or for company use provided that IJCAI copyright and the source are indicated, and that the copies are not used in a way that implies IJCAI endorsement of a product or service of an employer, and that the copies per se are not offered for sale. The foregoing right shall not permit the posting of the paper in electronic or digital form on any computer network, except by the author or the author's employer, and then only on the author's or the employer's own World Wide Web page or ftp site. Such Web page or ftp site, in addition to the aforementioned requirements of this Paragraph, must provide an electronic reference or link back to the IJCAI electronic server (http://www.ijcai.org), and shall not post other IJCAI copyrighted materials not of the author's or the employer's creation (including tables of contents with links to other papers) without IJCAI's written permission;
5. make limited distribution of all or portions of the above paper prior to publication.
6. In the case of work performed under U.S. Government contract, IJCAI grants the U.S. Government royalty-free permission to reproduce all or portions of the above paper, and to authorize others to do so, for U.S. Government purposes. In the event the above paper is not accepted and published by IJCAI, or is withdrawn by the author(s) before acceptance by IJCAI, this agreement becomes null and void.

Author's (or Employer's Representative) Signature	Date
Employer for whom work was performed	Title (if not author)

Clear Form

CITED LITERATURE

Abe, N., Zadrozny, B., and Langford, J.: An iterative method for multi-class cost-sensitive learning. In KDD, pages 3–11. ACM, 2004.

Andersen, M., Dahl, J., and Vandenberghe, L.: Cvxopt, python software for convex optimization, 2019.

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

Asif, K., Zhang, L., Derrible, S., Indacochea, E., Ozevin, D., and Ziebart, B. D.: Using machine learning to predict welding quality from airborne ae sensor data. 9th International Conference on Acoustic Emission (ICAE), 2019. Manuscript submitted.

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

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

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

Branson, S., Beijbom, O., and Belongie, S.: Efficient large-scale structured learning. In Computer Vision and Pattern Recognition, pages 1806–1813. IEEE, 2013.

Brefeld, U., Geibel, P., and Wysotzki, F.: Support vector machines with example dependent costs. In ECML, pages 23–34. Springer, 2003.

Breiman, L.: Bagging predictors. Machine learning, 24(2):123–140, 1996.

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

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

Chung, Y.-A., Lin, H.-T., and Yang, S.-W.: Cost-aware pre-training for multiclass cost-sensitive deep learning. arXiv preprint arXiv:1511.09337, 2015.

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

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.

Dalvi, N., Domingos, P., Sanghai, S., Verma, D., et al.: Adversarial classification. In KDD, pages 99–108. ACM, 2004.

Davis, J. V., Ha, J., Rossbach, C. J., Ramadan, H. E., and Witchel, E.: Cost-sensitive decision tree learning for forensic classification. In ECML, pages 622–629. Springer, 2006.

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

Elkan, C.: The foundations of cost-sensitive learning. In IJCAI, pages 973–978, 2001.

Fan, W., Stolfo, S. J., Zhang, J., and Chan, P. K.: Adacost: misclassification cost-sensitive boosting. In ICML, pages 97–105, 1999.

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, 2018.

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

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

Ferguson, T. S.: Game Theory, Second Edition. 2014.

Finley, T. and Joachims, T.: SVMpython, 2007.

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.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y.: Generative adversarial nets. In Advances in neural information processing systems, pages 2672–2680, 2014.

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.

Gurobi Optimization, I.: Gurobi optimizer reference manual, 2015.

Hoai, M. and De la Torre, F.: Max-margin early event detectors. International Journal of Computer Vision, 107(2):191–202, 2014.

Höffgen, K.-U. and Simon, H. U.: Robust trainability of single neurons. In Proceedings of the fifth annual workshop on Computational learning theory, pages 428–439. ACM, 1992.

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.

Joachims, T.: Making large scale SVM learning practical. Technical report, Universität Dortmund, 1999.

Khan, S. H., Hayat, M., Bennamoun, M., Sohel, F. A., and Togneri, R.: Cost-sensitive learning of deep feature representations from imbalanced data. IEEE transactions on neural networks and learning systems, 29(8):3573–3587, 2018.

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

Knoll, U., Nakhaeizadeh, G., and Tausend, B.: Cost-sensitive pruning of decision trees. In ECML, pages 383–386. Springer, 1994.

Koppula, H. S. and Saxena, A.: Anticipating human activities using object affordances for reactive robotic response. IEEE transactions on pattern analysis and machine intelligence, 38(1):14–29, 2016.

Kumar, V., Albert, S., Chandrasekhar, N., and Jayapandian, J.: Evaluation of welding skill using probability density distributions and neural network analysis. Measurement, 116:114–121, 2018.

Lafferty, J., McCallum, A., and Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the eighteenth international conference on machine learning, ICML, volume 1, pages 282–289, 2001.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

Lanckriet, G. R., Ghaoui, L. E., Bhattacharyya, C., and Jordan, M. I.: A robust minimax approach to classification. JMLR, 3:555–582, 2003.

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*, J., Asif*, K., Wang, H., Ziebart, B. D., and Berger-Wolf, T.: Adversarial sequence tagging. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, pages 1690–1696. AAAI Press, 2016.

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

Lin, H.-T.: A simple cost-sensitive multiclass classification algorithm using one-versus-one comparisons. National Taiwan University, Tech. Rep, 2010.

Ling, C. X., Yang, Q., Wang, J., and Zhang, S.: Decision trees with minimal costs. In ICML, pages 544–551. ACM, 2004.

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

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

Margineantu, D. D.: Class probability estimation and cost-sensitive classification decisions. In ECML, pages 270–281. Springer, 2002.

McCallum, A., Freitag, D., and Pereira, F.: Maximum entropy Markov models for information extraction and segmentation. In Proc. International Conference on Machine Learning, pages 591–598, 2000.

McMahan, H. B., Gordon, G. J., and Blum, A.: Planning in the presence of cost functions controlled by an adversary. In Proceedings of the International Conference on Machine Learning, pages 536–543, 2003.

Qin, Z., Wang, A. T., Zhang, C., and Zhang, S.: Cost-sensitive classification with k-nearest neighbors. In Knowledge Science, Engineering and Management, pages 112–131. Springer, 2013.

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

Reyes-Ortiz, J.-L., Oneto, L., Samà, A., Parra, X., and Anguita, D.: Transition-aware human activity recognition using smartphones. Neurocomputing, 171:754–767, 2016.

Reyes Ortiz, J. L., Oneto, L., Samà Monsonís, A., Ghio, A., Llanas Parra, X., and Anguita, D.: Transition-aware human activity recognition using smartphones. Neurocomputing, 2015.

Sarawagi, S. and Cohen, W. W.: Semi-markov conditional random fields for information extraction. In Advances in Neural Information Processing Systems, pages 1185–1192, 2004.

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

Sha, F. and Pereira, F.: Shallow parsing with conditional random fields. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, pages 134–141, 2003.

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

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

Song, H.-J., Son, J.-W., Noh, T.-G., Park, S.-B., and Lee, S.-J.: A cost sensitive part-of-speech tagging: differentiating serious errors from minor errors. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, pages 1025–1034. Association for Computational Linguistics, 2012.

Sumesh, A., Rameshkumar, K., Mohandas, K., and Babu, R. S.: Use of machine learning algorithms for weld quality monitoring using acoustic signature. Procedia Computer Science, 50:316–322, 2015.

Sumesh, A., Thekkuden, D. T., Nair, B. B., Rameshkumar, K., and Mohandas, K.: Acoustic signature based weld quality monitoring for smaw process using data mining algorithms. In Applied Mechanics and Materials, volume 813, pages 1104–1113. Trans Tech Publ, 2015.

Tang, Y.: Deep learning using linear support vector machines. arXiv preprint arXiv:1306.0239, 2013.

Ting, K. M.: A comparative study of cost-sensitive boosting algorithms. In ICML, 2000.

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

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

Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y.: Large margin methods for structured and interdependent output variables. In JMLR, pages 1453–1484, 2005.

Tu, H.-H. and Lin, H.-T.: One-sided support vector regression for multiclass cost-sensitive classification. In ICML, volume 2, page 5, 2010.

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

Turney, P. D.: Types of cost in inductive concept learning. arXiv preprint cs/0212034, 2002.

Viterbi, A. J.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. Information Theory, IEEE Transactions on, 13(2):260–269, 1967.

von Neumann, J. and Morgenstern, O.: Theory of Games and Economic Behavior. Princeton University Press, 1947.

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

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

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

Wang, R. and Tang, K.: Minimax classifier for uncertain costs. arXiv preprint arXiv:1205.0406, 2012.

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

Xing, Z., Pei, J., Dong, G., and Philip, S. Y.: Mining sequence classifiers for early prediction. In SDM, pages 644–655. SIAM, 2008.

Zadrozny, B., Langford, J., and Abe, N.: Cost-sensitive learning by cost-proportionate example weighting. In ICDM, pages 435–442, 2003.

Zhang, L., Abbasi, Z., Yuhas, D., Ozevin, D., Indacochea, E., et al.: Real-time nondestructive monitoring of the gas tungsten arc welding (gtaw) process by combined airborne acoustic emission and non-contact ultrasonics. In Nondestructive Characterization and Monitoring of Advanced Materials, Aerospace, Civil Infrastructure, and Transportation XII, volume 10599, page 105991V. International Society for Optics and Photonics, 2018.

Zhou, Z.-H. and Liu, X.-Y.: Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Transactions on Knowledge & Data Engineering, (1):63–77, 2006.

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

VITA

NAME Kaiser Newaj Asif

EDUCATION B.S., Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, 2009

PUBLICATIONS Kaiser Asif, Wei Xing, Sima Behpour, and Brian D. Ziebart, “Adversarial Cost-Sensitive Classification”, In Proceedings of the International Conference on Uncertainty in Artificial Intelligence (UAI), 2015.

Jia Li*, Kaiser Asif*, Hong Wang, Brian D. Ziebart, and Tanya Berger-Wolf, “Adversarial Sequence Tagging”, In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 2016.

Rizal Fathony, Kaiser Asif, Anqi Liu, Mohammad Ali Bashiri, Wei Xing, Sima Behpour, Xinhua Zhang, and Brian D Ziebart, “Consistent Robust Adversarial Prediction for General Multiclass Classification”, arXiv preprint arXiv:1812.07526, 2018.

Rizal Fathony, Anqi Liu, Kaiser Asif, and Brian Ziebart, “Adversarial Multiclass Classification: A Risk Minimization Perspective”, In Proceedings of the Advances in Neural Information Processing Systems (NIPS), 2016.

Hong Wang, Wei Xing, Kaiser Asif, and Brian D. Ziebart, Adversarial Prediction Games for Multivariate Losses, Advances in Neural Information Processing Systems (NIPS), 2015.

Kaiser Asif and Brian D. Ziebart, ”Inferring and Learning Subgoal Sequences”, International Conference on Machine Learning workshop on Robot Learning (ICML workshop), 2013.