Robust Cost-Sensitive Confidence-Weighted Classification

Alnur Ali
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA, U.S.A. 15213
Email: alnurali@cmu.edu

Kevyn Collins-Thompson
School of Information
University of Michigan
Ann Arbor, MI, U.S.A. 48109
Email: kevynct@umich.edu

***Abstract*—We introduce confidence-weighted (CW) online learning algorithms for robust, cost-sensitive classification. Our work extends the original confidence-weighted optimization framework in two important directions. First, we show how the original *value at risk (VaR)* probabilistic constraint in CW algorithms can be generalized to a worst-case *conditional value at risk (CVaR)* constraint for more robust learning from cost-weighted examples. Second, we show how to reduce adversarial feature noise, which can be useful in fraud detection scenarios, by reframing the optimization problem in terms of maximum a posteriori estimation. The resulting optimization problems can be solved efficiently. Experiments on real-world and synthetic datasets show that our robust, cost-sensitive extensions consistently reduce the cost incurred in both online and batch learning settings. We also demonstrate a correspondence between the VaR and CVaR constraints used for classification and *uncertainty sets* used in robust optimization, leading toward a rich family of potential extensions to CW algorithms.**

I. Introduction

Online learning algorithms are an appealing choice for many classification problems, given their ability to handle large datasets efficiently and adapt to evolving patterns over time. Furthermore, many important classification scenarios, such as fraud detection and medical diagnosis, are *cost-sensitive*, e.g. where the cost of a false negative is much higher than a false positive. Traditional classifiers, however, are not sensitive to this asymmetry and typically make poor decisions in cost-sensitive situations, resulting in higher overall costs for users or systems. In particular, confidence weighted (CW) linear classification is a recently introduced online learning algorithm that achieves excellent performance on a variety of classification tasks [1], [2], but in its standard form does not handle cost-sensitive scenarios, or tasks where more sensitive measures of misclassification risk are important.

In this paper, we introduce a family of CW classification algorithms that support robust, cost-sensitive learning. To do this, we extend the standard online CW linear classification algorithm in two important directions. First, we show how to reformulate the *value at risk (VaR)* constraint in the original CW optimization problem to support a more general worst case *conditional value at risk (CVaR)* constraint. This constraint can be customized by the user to achieve the appropriate level of cost-sensitivity for the task at hand, and the resulting optimization problem can be solved efficiently. Second, we extend our algorithm to be robust to noise injected by an adversary into the input data. As part of our derivation, we show the correspondence between these VaR and CVaR constraints, and uncertainty sets used in robust optimization: a connection which enables the derivation of further extensions to CW algorithms. We validate our algorithm experimentally on real and synthetic data.

II. Related Work

There has been a great deal of work in the machine learning community on cost-sensitive learning: one paper that is relevant to our approach is [3], which shows that minimizing the cost-sensitive generalization error is equivalent to minimizing the weighted error on an observed sample, where the weights are proportional to the costs present in the sample. These weights may be integrated into a learning algorithm in a *white box* fashion, in which the internals of the algorithm are modified to accomodate the cost, or in a *black box* fashion, in which the algorithm is not modified. In this paper, our goal is to propose basic changes to the underlying objective of the standard CW algorithm, including parameterized generalizations, and so we use a white-box approach.

In portfolio theory there has, naturally, been much work on minimizing the risk of loss. The seminal paper of [4] proposed the *mean/variance* approach: a portfolio appropriately trading-off the expected return on the investments and the overall risk of loss is sought, with risk in this context being defined as the variance of the returns. [5] extend algorithms for online learning with expert advice to make use of the mean/variance approach. [6] study various aspects of a relatively more recent and sensitive measure of risk, known as conditional value at risk (CVaR) (or expected shortfall); this risk measure is a central part of our algorithm objective later in this paper.

These risk measures have, in turn, recently attracted the attention of machine learning researchers. To our knowledge the most similar related work to ours is that of [7], who adapted the perceptron to make use of CVaR for the purpose of cost-sensitive learning; Kashima's approach shares our high-level goal of robust, cost-sensitive classification, but uses a very different learning mechanism. While our approach uses a white-box, online, confidence-weighted learning framework, Kashima used a black-box, meta-learning framework that wraps existing batch-oriented classifiers.

The CVaR-based learning objective has also arisen in the work of [8], who showed that the E-ν support vector machine (SVM) [9] can be seen as minimizing a CVaR-based criterion;

they used this interpretation to provide a justification for the positive empirical performance of the E-ν SVM. In earlier work, [10] introduced the minimax probability machine, which implemented the value at risk measure (VaR). [11] derive a variant of the CW algorithm for the setting of *learning to trade*. Additionally, [12] introduce a variant of the CW algorithm known as adaptive regularization of weight vectors (AROW): we do not consider this algorithm in our paper, although our methods could be extended to it as well.

III. Robust Cost-Sensitive Confidence-Weighted Linear Classification

In this section, we derive a robust, cost-sensitive generalization of the standard CW algorithm.

We begin by reviewing the standard CW algorithm. In contrast to most algorithms for online learning, the CW algorithm maintains a Gaussian distribution, with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, over linear classifiers. On each round $t \in \{1, \ldots, T\}$, the algorithm receives an example, $\mathbf{x}_t$, predicts its label by computing $\operatorname{sgn}(\boldsymbol{\mu}^T \mathbf{x}_t)$, and then updates its estimates of $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ by solving the following optimization problem:

$$\begin{array}{ll} \underset{\boldsymbol{\mu},\boldsymbol{\Sigma}}{\text{minimize}} & D_{KL}(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \,||\, \mathcal{N}(\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1})) \quad (1) \\ \text{subject to} & P(y_t \mathbf{w}^T \mathbf{x}_t \geq 0) \geq \eta. \quad (2) \end{array}$$

Here, D_{KL} is the K-L divergence, $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ are the parameter estimates that we want to find on round t, $(\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1})$ are the parameter estimates at the end of round $t-1$, y_t is the label of example $\mathbf{x}_t$, $\mathbf{w}$ is a classifier drawn from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, P is a distribution over misclassifications, and η is the user-specified desired probability of a correct classification.

Eq. 1 is essentially a regularization term: it forces one to find a solution, $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, on round t, that does not stray too much from previous solutions. Eq. 2, on the other hand, can be thought of as a data fit term: it constrains the solution to classify the current example, $(\mathbf{x}_t, y_t)$, correctly, confidently, and with high probability.

The quantity $y_t \mathbf{w}^T \mathbf{x}_t$ in Eq. 2 is the *margin*, which quantifies the degree of misclassification; we denote the margin as M_t. In cost-sensitive scenarios, one may pay a price, $c \geq 0$, for these misclassifications; more generally, the price may be a function of the inputs, $c(\mathbf{x})$ [3]. Consequently, one may require that the learning algorithm suffer the loss $c(\mathbf{x})M_t$, instead of just M_t.

A. Example-level Conditional Value at Risk Constraint

Before we present our method for controlling the costs of misclassifications in the CW algorithm, we first discuss some techniques for minimizing potential losses that are used in portfolio optimization. In portfolio optimization, the goal is to find a portfolio striking an appropriate balance between the expected return on the investments and the overall risk of loss; a portfolio, $\mathbf{z}$, can be defined as a point on the $(D-1)$-dimensional probability simplex: $\mathbf{z} \in \mathbb{R}_+^D$, $\sum_{d=1}^D z_d = 1$, $0 \leq z_d \leq 1$, $\forall d$, with each component, z_d, representing the degree of investment in an asset. The portfolio loss, R, can, in general, be defined as a function of the portfolio and the random vector representing the returns, $\mathbf{r}$: $R = f(\mathbf{z}, \mathbf{r})$[1].

Many risk measures exist in portfolio theory; one is value at risk (VaR) [13], which is defined as the loss below which the worst $\epsilon\%$ of losses lie, where $\epsilon = 1-\eta$. ϵ is usually set to a small number, such as 0.05. This is equivalent to the ϵ-quantile of R, which can be written as

$$\mathrm{VaR}_\epsilon(R) = \min_l P(R \leq l) \leq \epsilon$$

where P is the distribution over losses. If we rewrite Eq. 2 as $P(M_t \leq 0) \leq \epsilon$, then we can see that this is equivalent to requiring $\mathrm{VaR}_\epsilon(M_t) = 0$.

An alternative risk measure is conditional value at risk (CVaR), which is defined as the mean of the worst $\epsilon\%$ of losses; this is equivalent to the mean of the ϵ-tail distribution of R [14], which can be written as

$$\begin{aligned} \mathrm{CVaR}_\epsilon(R) &= \mathrm{E}_R[R < r^*] \\ &= r^* - \frac{1}{\epsilon}\mathrm{E}_R[(r^* - R)_+] \end{aligned}$$

where $r^* = \mathrm{VaR}_\epsilon(R)$, and $(z)_+ = \max(0, z)$. Unlike VaR, CVaR is sensitive to the shape of the distribution tail, and thus can recognize scenarios where the worst losses are skewed toward catastrophic outcomes. This can be a useful property when trying to control costs [6].

CVaR can be computed by a Monte Carlo approximation for arbitrary distributions over losses, P [6]; however, this approach can be problematic in settings in which a rapid response is essential, such as online learning. If we instead make some weak assumptions on the nature of P, then Prop. 2 in [15] allows us to compute CVaR in closed form, under the assumption of a Gaussian distribution over losses, and also in the worst case, under the increasingly non-commital assumptions of a symmetric, symmetric and unimodal, and an arbitrary distribution, known only up to the first and second moments.

We replace Eq. 2 in the standard CW optimization problem with the cost-sensitive margin $c(\mathbf{x}_t)M_t$, and a CVaR risk measure over this random variable; this leads to the following optimization problem:

$$\begin{array}{ll} \underset{\boldsymbol{\mu},\boldsymbol{\Sigma}}{\text{minimize}} & D_{KL}(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \,||\, \mathcal{N}(\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1})) \quad (3) \\ \text{subject to} & \mathrm{CVaR}_\epsilon(c(\mathbf{x}_t)M_t) \geq 0. \quad (4) \end{array}$$

Next, we solve this problem under the assumption that the model parameters are modeled with a Gaussian distribution, with the distribution of cost-sensitive margin losses also being Gaussian; other solutions under alternative distributional assumptions for the margin loss, such as might result from the nature of the data distribution, follow similarly by using the results from Table I as described later.

Eq. 3 is the K-L divergence between two Gaussians, which can be written as

$$\begin{aligned} &\frac{1}{2}(\log(\frac{\det \boldsymbol{\Sigma}_{t-1}}{\det \boldsymbol{\Sigma}}) + \mathrm{trace}(\boldsymbol{\Sigma}_{t-1}^{-1}\boldsymbol{\Sigma}) + \quad (5) \\ &(\boldsymbol{\mu}_{t-1} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}_{t-1}(\boldsymbol{\mu}_{t-1} - \boldsymbol{\mu}) - D) \end{aligned}$$

where D is the dimensionality of $\boldsymbol{\mu}$. Let us denote the mean of the margin as $\bar{M}_t = y_t \boldsymbol{\mu}^T \mathbf{x}_t$, and its variance as $v_t = \mathbf{x}_t^T \boldsymbol{\Sigma} \mathbf{x}_t$.

[1] In this paper, we treat losses as negative numbers.

Applying Prop. 2, Eq. 22 in [15], with $f \geq 0$, $\mu_x = c(\mathbf{x}_t)\bar{M}_t$, and $\sigma_x = \sqrt{v_t}$, allows us to write Eq. 4 as

$$\frac{1}{\sqrt{2\pi}\epsilon}\exp(-\frac{1}{2}(\Phi^{-1}(1-\epsilon))^2)\sigma_x^2 - c(\mathbf{x}_t)\bar{M}_t \leq 0 \qquad (6)$$

where $\Phi^{-1}(\cdot)$ is the standard normal inverse cumulative distribution function; we have squared σ_x in Eq. 6 in order to make this constraint convex in $\mathbf{\Sigma}$.

Putting Eqs. 5 and 6 together yields the following revised convex optimization problem:

$$\begin{array}{ll} \underset{\boldsymbol{\mu},\mathbf{\Sigma}}{\text{minimize}} & \frac{1}{2}(\log(\frac{\det \mathbf{\Sigma}_{t-1}}{\det \mathbf{\Sigma}}) + \text{trace}(\mathbf{\Sigma}_{t-1}^{-1}\mathbf{\Sigma}) + \\ & (\boldsymbol{\mu}_{t-1} - \boldsymbol{\mu})^T \mathbf{\Sigma}_{t-1}(\boldsymbol{\mu}_{t-1} - \boldsymbol{\mu}) - D) \\ \text{subject to} & \omega_\epsilon v_t - c(\mathbf{x}_t)\bar{M}_{t-1} \leq 0 \end{array} \qquad (7)$$

where we have defined $\omega_\epsilon = \frac{1}{\sqrt{2\pi}\epsilon}\exp(-\frac{1}{2}(\Phi^{-1}(1-\epsilon))^2)$. The Lagrangian of Eq. 7 is

$$\begin{array}{rl} L(\boldsymbol{\mu}, \mathbf{\Sigma}, \lambda) = & \frac{1}{2}(\log(\frac{\det \mathbf{\Sigma}_{t-1}}{\det \mathbf{\Sigma}}) + \text{trace}(\mathbf{\Sigma}_{t-1}^{-1}\mathbf{\Sigma}) + \\ & (\boldsymbol{\mu}_{t-1} - \boldsymbol{\mu})^T \mathbf{\Sigma}_{t-1}(\boldsymbol{\mu}_{t-1} - \boldsymbol{\mu}) - D) + \\ & \lambda(\omega_\epsilon v_t - c(\mathbf{x}_t)\bar{M}_t). \end{array} \qquad (8)$$

Solving Eq. 8 for the optimal $\boldsymbol{\mu}$ yields

$$\boldsymbol{\mu} = \boldsymbol{\mu}_{t-1} + \lambda c(\mathbf{x}_t) y_t \mathbf{\Sigma}_{t-1}\mathbf{x}_t. \qquad (9)$$

Solving Eq. 8 for the optimal $\mathbf{\Sigma}^{-1}$ yields

$$\mathbf{\Sigma}^{-1} = \mathbf{\Sigma}_{t-1}^{-1} + 2\lambda\omega_\epsilon \mathbf{x}_t \mathbf{x}_t^T$$

which can be inverted using the Sherman-Morrison-Woodbury identity:

$$\mathbf{\Sigma} = \mathbf{\Sigma}_{t-1} - \mathbf{\Sigma}_{t-1}\mathbf{x}_t \frac{2\lambda\omega_\epsilon}{1 + 2\lambda\omega_\epsilon v_{t-1}} \mathbf{x}_t^T \mathbf{\Sigma}_{t-1}. \qquad (10)$$

Plugging Eqs. 9 and 10 back into Eq. 8, setting this quantity equal to zero, rearranging, and omitting terms without a dependence on λ yields

$$c(\mathbf{x}_t)\bar{M}_{t-1} + c(\mathbf{x}_t)^2 \lambda v_{t-1} = \omega_\epsilon v_{t-1} - \omega_\epsilon v_{t-1}^2 \frac{2\lambda\omega_\epsilon}{1 + 2\lambda\omega_\epsilon v_{t-1}} \qquad (11)$$

which is quadratic in λ. Complementary slackness implies $\lambda \geq 0$, and hence the optimal λ is

$$\lambda = \max(0, \psi) \qquad (12)$$

where ψ is the positive root of Eq. 11 equal to

$$\begin{array}{rl} \psi = & \frac{1}{4\omega_\epsilon c(\mathbf{x}_t) v_{t-1}}(-(c(\mathbf{x}_t) + 2\omega_\epsilon \bar{M}_{t-1}) + \\ & \sqrt{(c(\mathbf{x}_t) + 2\omega_\epsilon \bar{M}_{t-1})^2 - 8\omega_\epsilon(c(\mathbf{x}_t)\bar{M}_{t-1} - \omega_\epsilon v_{t-1})}). \end{array}$$

To summarize, the solution to the optimization problem in Eqs. 3, 4 can be obtained by Eqs. 12, 9, and 10. If we make alternate assumptions on the margin distribution, then only ω_ϵ changes; Table I presents the value of ω_ϵ for different margin distribution assumptions, based on results from [15].

This completes our derivation of a CVaR-based chance constraint for cost-sensitive CW classification. In the next section, we give an additional refinement whose goal is to increase robustness of the classifier by estimating the empirical distribution over example costs in the data.

B. Cost-Sensitive Tolerance for Misclassification

Fig. 1 plots the value of the Lagrange multiplier λ computed by Eq. 12 as a function of the desired misclassification probability, ϵ, and the example cost level, $c(\mathbf{x})$, under the assumption of a Gaussian margin distribution; for simplicity, we set $\bar{M}_t = 0$ and $v_t = 1$. Across a variety of cost levels, we can see that smaller values of ϵ lead to larger values of λ, which may yield overly aggressive updates to $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$.

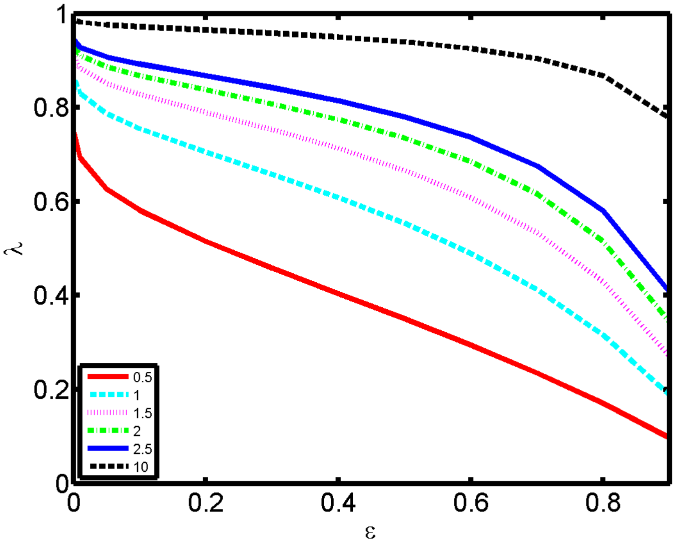


Figure 1. Sensitivity of the Lagrange multiplier λ as a function of the desired misclassification probability, ϵ, across several different example cost levels, $c(\mathbf{x})$, showing one curve per choice of $c = 0.5, \ldots, 10$, under the assumption of a Gaussian margin distribution.

In general, we want to ensure that the desired probability of misclassification ϵ is connected to the expected empirical cost of misclassification: if an example is very costly compared to the overall population, tolerance for misclassification should be low (small ϵ), whereas average-cost examples may allow higher tolerance for misclassification (higher ϵ). To implement this idea, we introduce a history buffer B_t for estimating the empirical distribution over example costs.

With this empirical distribution, we can identify when an individual example is relatively high cost, or not, by making ϵ depend on this relative cost. In this study we define a simple two-level function defined by Eq. 13. The result is that to correct the most costly mistakes, we only make updates to λ, $\boldsymbol{\mu}$, and $\mathbf{\Sigma}$ using a small desired misclassification probability α for examples with the highest costs, $c(\mathbf{x})$. We use a second, larger, allowed misclassification probability, β, to make updates on examples with lower costs. More general definitions of Eq. 13 are certainly possible and a subject of future work. In our current algorithm below, as learning progresses, we accrue the example costs, $c(\mathbf{x}_t)$, in the buffer, B_t; we then compute $c^* = \text{CVaR}_\tau(B_t)$ over the buffer, where τ specifies the percentile of the cost distribution that is to be considered highest relative cost. The desired misclassification probability ϵ is then as follows:

$$\epsilon = \left\{ \begin{array}{ll} \alpha & \text{if } c^* \leq c(\mathbf{x}_t) \\ \beta & \text{otherwise.} \end{array} \right. \qquad (13)$$

Table I. VALUES OF UPDATE FACTOR ω_ϵ FOR VARIOUS MARGIN DISTRIBUTION ASSUMPTIONS.

MARGIN DISTRIBUTION ASSUMPTION	ω_ϵ
GAUSSIAN	$1/\epsilon\sqrt{2\pi} \cdot \exp\left(-\frac{1}{2}(\Phi^{-1}(1-\epsilon))^2\right)$
SYMMETRIC	$1/\sqrt{2\epsilon}$ FOR $\epsilon \in (0, 1/2]$; $1/\sqrt{2}\epsilon \cdot \sqrt{1-\epsilon}$ FOR $\epsilon \in [1/2, 1)$
SYMMETRIC AND UNIMODAL	$2/3\sqrt{\epsilon}$ FOR $\epsilon \in (0, 1/3]$; $\sqrt{3}(1-\epsilon)$ FOR $\epsilon \in [1/3, 2/3]$; $1/3\epsilon \cdot 2\sqrt{1-\epsilon}$ FOR $\epsilon \in [2/3, 1)$
ARBITRARY	$\sqrt{1-\epsilon}/\sqrt{\epsilon}$

We then update λ, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$, based on ω_ϵ, as in Eqs. 12, 9, and 10.

We refer to the algorithm making use of the update rules in Eqs. 13, 12, 9, and 10 as CW-CVaR; this algorithm is summarized in Alg. 1.

Algorithm 1 CW-CVaR.

```
Input: desired misclassification probability for highest-cost examples, $\alpha$; for lowest-cost examples, $\beta$; cost-sensitive misclassification tolerance $\tau$.
Initialize: buffer $B_0 = \varnothing$; classifier parameters, $\boldsymbol{\mu}_0 = \mathbf{0}$, $\boldsymbol{\Sigma}_0 = \mathbf{I}$
for $t = 1$ to $T$ rounds do
    receive $(\mathbf{x}_t, y_t, c(\mathbf{x}_t))$, $y_t \in \{-1, 1\}$
    add $c(\mathbf{x}_t)$ to buffer: $B_t = B_{t-1} \cup c(\mathbf{x}_t)$
    compute $c^* = \text{CVaR}_\tau(B_t)$
    compute $\epsilon$ by Eq. 13, using $c^*$
    predict $\hat{y}_t = \text{sgn}(\boldsymbol{\mu}_{t-1}^T \mathbf{x}_t)$
    update:
        • compute $\lambda$ by Eq. 12, using $\omega_\epsilon$
        • $\boldsymbol{\mu}_t = \boldsymbol{\mu}_{t-1} + \lambda c(\mathbf{x}_t) y_t \boldsymbol{\Sigma}_{t-1} \mathbf{x}_t$ (Eq. 9)
        • $\boldsymbol{\Sigma}_t = \boldsymbol{\Sigma}_{t-1} - \boldsymbol{\Sigma}_{t-1} \mathbf{x}_t \frac{2\lambda\omega_\epsilon}{1+2\lambda\omega_\epsilon v_{t-1}} \mathbf{x}_t^T \boldsymbol{\Sigma}_{t-1}$ (Eq. 10)
end for
```

C. Sensitivity of Classifier Updates to Margin Distribution Assumptions

Fig. 2 plots the value of the Lagrange multiplier λ as a function of the example cost level, $c(\mathbf{x})$, and the desired misclassification probability, ϵ, under different assumptions on the margin distribution; we again set $\bar{M}_t = 0$ and $v_t = 1$. We can see that for high-cost examples, there is little difference between the value of λ for the most restrictive distribution assumption (Gaussian) and the least restrictive (arbitrary). However, as the cost level decreases, the ratio curves show a more pronounced difference, especially for lower desired misclassification probabilities. Thus, we may surmise that although the updates λ, and consequently to $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, are sensitive to the margin distribution assumption, this sensitivity is diminished at higher cost levels.

IV. ROBUSTNESS TO ADVERSARIAL FEATURE NOISE

In certain cost-sensitive scenarios, one may wish to be resistant to noise injected by an adversary into the input data: for instance, in a loan approval scenario, an adversary may falsify parts of their loan application in order to deceive the system into approving their loan. In this section, we further extend CW-CVaR to be robust to this kind of noise.

We assume that we are given some prior knowledge of the features x_d that might be corrupted; we encode this knowledge in a prior distribution $P(\boldsymbol{\mu}|\boldsymbol{\nu}, \boldsymbol{\Upsilon})$, with mean $\boldsymbol{\nu}$ and covariance $\boldsymbol{\Upsilon}$, over $\boldsymbol{\mu}$: if a feature x_d is suspected to be corrupt, then its corresponding weight in the prior mean, ν_d, can be set to a diminished value.

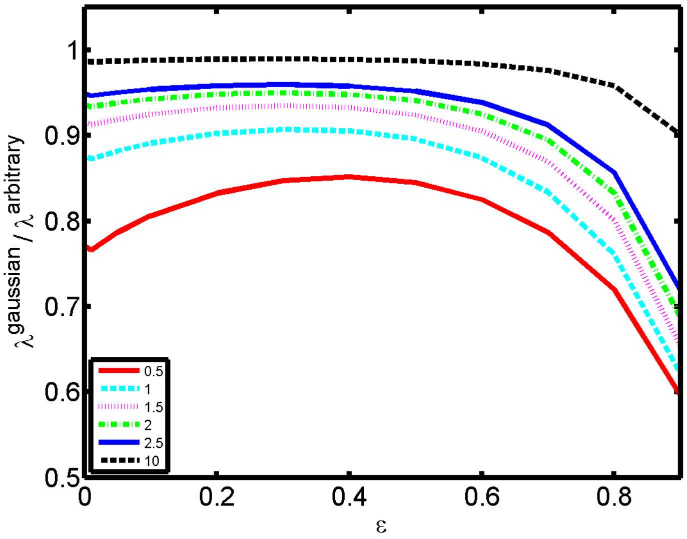


Figure 2. Sensitivity of the ratio $\lambda^{\text{gaussian}}/\lambda^{\text{arbitrary}}$ as a function of desired misclassification probability ϵ, showing one curve per choice of $c = 0.5, \ldots, 10$. The curves demonstrate the effect on λ of assuming a Gaussian assumption over margin losses, compared to assuming an arbitrary distribution, and how this difference varies with cost-sensitivity.

Multiplying the prior by the likelihood, $P(\mathbf{w}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$, and normalizing yields the posterior:

$$P(\boldsymbol{\mu}|\mathbf{w}, \boldsymbol{\nu}, \boldsymbol{\Sigma}, \boldsymbol{\Upsilon}) = \frac{1}{Z} P(\mathbf{w}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) P(\boldsymbol{\mu}|\boldsymbol{\nu}, \boldsymbol{\Upsilon})$$

where Z is a normalization constant.

If we assume that the prior is Gaussian, then, by conjugacy, the posterior will also be Gaussian, with mean $\tilde{\boldsymbol{\mu}}$ and covariance $\tilde{\boldsymbol{\Sigma}}$, where

$$\tilde{\boldsymbol{\Sigma}} = (\boldsymbol{\Upsilon}^{-1} + t\boldsymbol{\Sigma}^{-1})^{-1} \qquad (14)$$

$$\tilde{\boldsymbol{\mu}} = \tilde{\boldsymbol{\Sigma}}(\boldsymbol{\Upsilon}^{-1}\boldsymbol{\nu} + t\boldsymbol{\Sigma}^{-1}\mathbf{w}). \qquad (15)$$

We set $\mathbf{w} = \boldsymbol{\mu}$, which can be estimated via Alg. 1.

$\tilde{\boldsymbol{\mu}}$ is essentially an average of $\boldsymbol{\nu}$ and $\boldsymbol{\mu}$, weighted by the respective covariances $\boldsymbol{\Upsilon}$ and $\boldsymbol{\Sigma}$. As $t \to \infty$, and $\boldsymbol{\Upsilon}_{dd}/\boldsymbol{\Sigma}_{dd} \to \infty$ (assuming that $\boldsymbol{\Upsilon}$ and $\boldsymbol{\Sigma}$ are diagonal), $\tilde{\mu}_d$ will be increasingly dominated by μ_d, as opposed to ν_d: in other words, as more data is seen, the effect of the prior can be diminished.

We refer to the algorithm which first estimates $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ as in Alg. 1, then uses these estimates to compute $\tilde{\boldsymbol{\mu}}$ and $\tilde{\boldsymbol{\Sigma}}$

as in Eqs. 15 and 14, and finally uses $\tilde{\boldsymbol{\mu}}$ to make predictions, as CW-robust; this algorithm is summarized in Alg. 2.

Algorithm 2 CW-robust.

```
**Input:** desired misclassification probability for highest-cost examples, $\alpha$; for lowest-cost examples, $\beta$; cost-sensitive misclassification tolerance $\tau$; prior mean $\boldsymbol{\nu}$; prior covariance $\boldsymbol{\Upsilon}$;
**Initialize:** buffer $B_0 = \varnothing$; classifier parameters, $\boldsymbol{\mu}_0 = \mathbf{0}$, $\boldsymbol{\Sigma}_0 = \mathbf{I}$
**for** $t = 1$ **to** $T$ rounds **do**
    receive $(\mathbf{x}_t, y_t, c(\mathbf{x}_t))$, $y_t \in \{-1, 1\}$
    compute $B_t$, $c^*$, $\epsilon$, $\lambda$, $\boldsymbol{\mu}_t$, and $\boldsymbol{\Sigma}_t$ as in Alg. 1
    predict $\hat{y}_t = \operatorname{sgn}(\tilde{\boldsymbol{\mu}}_{t-1}^T \mathbf{x}_t)$
    update:
    • $\tilde{\boldsymbol{\Sigma}}_t = (\boldsymbol{\Upsilon}^{-1} + t\boldsymbol{\Sigma}_t^{-1})^{-1}$
    • $\tilde{\boldsymbol{\mu}}_t = \tilde{\boldsymbol{\Sigma}}_t(\boldsymbol{\Upsilon}^{-1}\boldsymbol{\nu} + t\boldsymbol{\Sigma}_t^{-1}\boldsymbol{\mu}_t)$
**end for**
```

V. Connections to Robust Optimization

In this section, we demonstrate a correspondence between the risk measures used in the standard CW and CW-CVaR optimization problems and uncertainty sets. Our purpose in including these results here is two-fold: first, to provide an alternate perspective on our CVaR robust optimization problem; and second, to suggest a mechanism that may enable users who wish to incorporate prior knowledge on $\boldsymbol{\mu}$ to derive a corresponding risk measure for use in the CW optimization problem, and vice-versa. [16] have used a similar approach to derive new robust algorithms for portfolio optimization.

Thm. 1 shows the VaR constraint in the CW optimization problem is equivalent to a constraint that places an ellipsoidal uncertainty set around $\boldsymbol{\mu}$.

Theorem 1. *Eq. 2 in the standard CW optimization problem is equivalent to enforcing a deterministic margin constraint for all values of $\tilde{\mathbf{w}} \in \mathcal{E}$, where $\mathcal{E}$ is an ellipsoid centered around $\boldsymbol{\mu}$.*

Proof: Assume that, instead of Eq. 2, we want the following deterministic constraint to hold instead: $y_t \tilde{\mathbf{w}}^T \mathbf{x}_t \geq 0$, $\forall \tilde{\mathbf{w}} \in \mathcal{E}$, where $\mathcal{E}$ is an ellipsoid centered around $\boldsymbol{\mu}$. This constraint is equivalent to

$$\begin{aligned} & \min_{||\mathbf{u}||_2 \leq 1} y_t(\boldsymbol{\mu} - \mathbf{Q}\mathbf{u})^T \mathbf{x}_t \\ = \quad & y_t \boldsymbol{\mu}^T \mathbf{x}_t - y_t \max_{||\mathbf{u}||_2 \leq 1} \mathbf{u}^T \mathbf{Q}^T \mathbf{x}_t \\ = \quad & y_t \boldsymbol{\mu}^T \mathbf{x}_t - y_t ||\mathbf{Q}^T \mathbf{x}_t||_2 \end{aligned} \tag{16}$$

where $\mathbf{Q} \in \mathbb{R}^{d \times d}$. If $\mathbf{Q} = \Phi^{-1}(\eta)/y_t \cdot \boldsymbol{\Sigma}\mathbf{x}_t$, and we omit the square root in Eq. 16, then this constraint is equivalent to Eq. 2. ■

Thm. 2 shows the CVaR constraint in the CW-CVaR optimization problem is equivalent to a constraint that places a polyhedral uncertainty set around $\mathbf{w}$.

Theorem 2. *Eq. 4 in the revised CW-CVaR optimization problem is equivalent to enforcing a deterministic margin constraint for all values of $\tilde{\mathbf{w}} \in \mathcal{P}$, where $\mathcal{P}$ is a polyhedron defined by $\mathbf{a}_i^T \tilde{\mathbf{w}} \leq b_i$, $\forall i \in \{1, \ldots, I\}$, $\forall i \in \{1, \ldots, I\}$, $\tilde{w}_d \geq 0$, $\forall d \in \{1, \ldots, D\}$.*

Proof: First, assume that, w.l.o.g., $c(\mathbf{x}_t) = 1$. Next, assume that instead of Eq. 2, we want the following deterministic constraint to hold: $y_t \tilde{\mathbf{w}}^T \mathbf{x}_t \geq 0$, $\forall \tilde{\mathbf{w}} \in \mathcal{P}$, where $\mathcal{P}$ is a polyhedron defined by $\mathbf{a}_i^T \tilde{\mathbf{w}} \leq b_i$, $\forall i \in \{1, \ldots, I\}$, $\tilde{w}_d \geq 0$, $\forall d \in \{1, \ldots, D\}$. This constraint is equivalent to

$$\begin{aligned} \min_{\tilde{\mathbf{w}}} \quad & y_t \tilde{\mathbf{w}}^T \mathbf{x}_t \\ \text{s.t.} \quad & \mathbf{a}_i^T \tilde{\mathbf{w}} \leq b_i, \ \forall i \in \{1, \ldots, I\} \\ & \tilde{w}_d \geq 0, \ \forall d \in \{1, \ldots, D\} \end{aligned} \tag{17}$$

The dual of Eq. 17 is

$$\begin{aligned} \max_{\lambda_i} \quad & \sum_{i=1}^{I} \lambda_i b_i \\ \text{s.t.} \quad & \sum_{i=1}^{I} \lambda_i a_{id} \leq y_t x_{td}, \ \forall d \in \{1, \ldots, D\}, \\ & \lambda_i \geq 0, \ \forall i \in \{1, \ldots, I\} \end{aligned} \tag{18}$$

$\text{CVaR}_\epsilon(M_t)$ can equivalently be written as [6]

$$\begin{aligned} \max_{\tilde{M}_t^{(s)}} \quad & M_t^* - \frac{1}{\epsilon} \sum_{s=1}^{S} p_s \tilde{M}_t^{(s)} \\ \text{s.t.} \quad & \tilde{M}_t^{(s)} \geq M_t^* - M_t^{(s)}, \ \forall s \in \{1, \ldots, S\} \\ & \tilde{M}_t^{(s)} \geq 0, \ \forall s \in \{1, \ldots, S\} \end{aligned} \tag{19}$$

where $M_t^* = \text{VaR}_\epsilon(M_t)$, $M_t^{(s)} = y_t \mathbf{w}^{(s)T} \mathbf{x}_t$, $\mathbf{w}^{(s)}$ are draws from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, and p_s is the probability of each draw.

The dual of Eq. 19 is

$$\begin{aligned} \min_{\lambda_s} \quad & \sum_{s=1}^{S} \lambda_s M_t^{(s)} \\ \text{s.t.} \quad & \sum_{s=1}^{S} \lambda_s \leq \frac{p_s}{\epsilon}, \ \forall s \in \{1, \ldots, S\} \\ & \lambda_s \geq 0, \ \forall s \in \{1, \ldots, S\}. \end{aligned} \tag{20}$$

If $a_{id} = \operatorname{sgn}(y_t x_{td}) y_t x_{td} \epsilon / p_s$, $b_i = -M_t^{(s)}$, $I = S$, and strong duality holds, then Eq. 17 is equivalent to Eq. 19. ■

In future work we intend to use the connection between risk measures and uncertainty sets to derive further adaptations of our online learning framework to specific problems.

VI. Experimental Evaluation

In this section, we describe our experimental results with the cost-sensitive CW-CVaR algorithm on two real cost-sensitive data sets, as well as with the adversarial noise resistant CW-robust algorithm on a synthetic data set.

A. Real Data Sets

We experimented with CW-CVaR under a variety of margin distribution assumptions, as well as with the standard CW algorithm. In order to understand the contributions of the example-level CVaR constraint described in Sec. III-A and the stream-level CVaR-based costs buffer described in Sec. III-B, we also experimented with the CW algorithm with the example costs, $c(\mathbf{x})$, incorporated into the update rules, which we denote as CW-costs, and CW-CVaR without the buffer, which we denote as CW-nobuffer.

We experimented with all these algorithms in an online learning setting, in which we tracked the cumulative cost incurred by each algorithm as it learns, as well as in a batch learning setting, in which we computed the cost incurred by each algorithm on a fixed test set after learning.

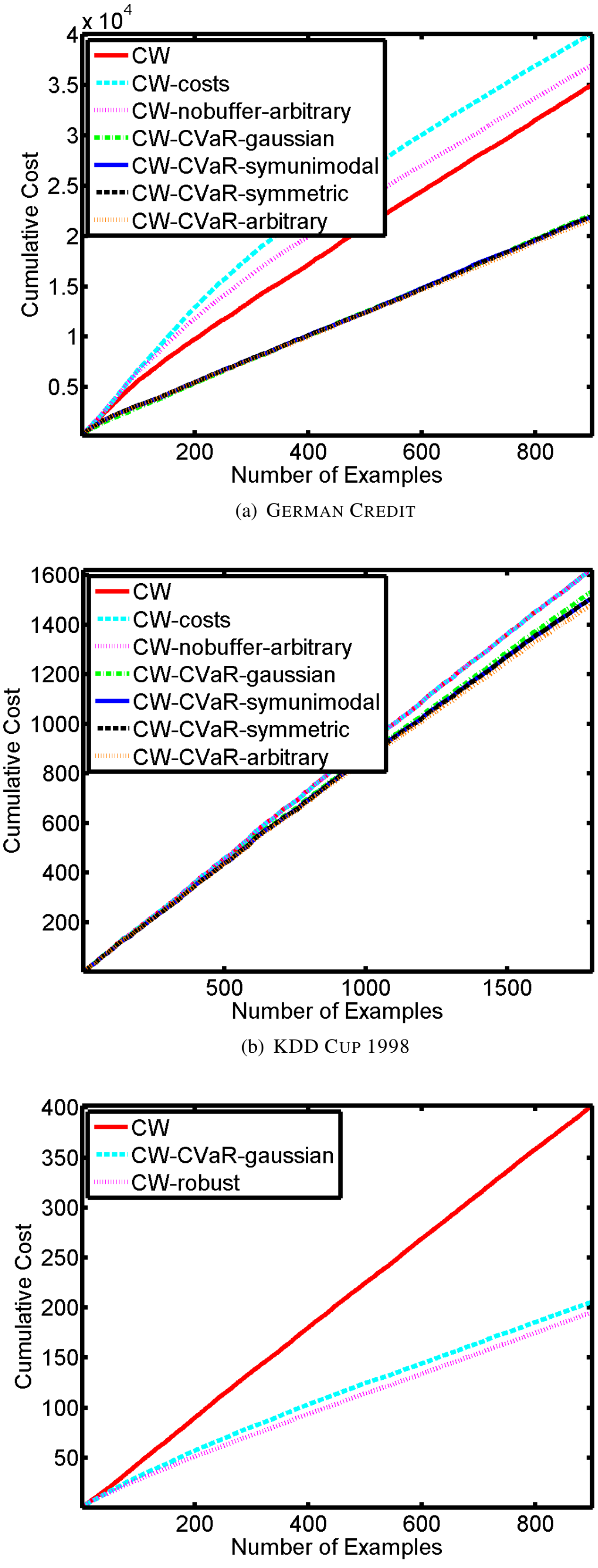
Each algorithm's hyper-parameters, ϵ, α, β, and the assumed margin distribution, were set by following the same protocol as in [1]: we fixed each hyper-parameter to the value[2] for which the algorithm incurred the least cumulative cost in the online learning setting on a single, randomly generated trial of the data. We used a fixed value of $\tau = 0.05$ for all experiments. All algorithms used a diagonal approximation to the full covariance matrix, $\mathbf{\Sigma}$, as described in [1].

We experimented with the cost-sensitive GERMAN CREDIT and KDD CUP 1998 data sets. Tables II and III present the cost incurred by each algorithm, and CVaR computed at $\epsilon = 0.05$ on each algorithm's empirical margin distribution on the test data, on these data sets, in the online and batch settings. The margin distribution assumption set by the hyper-parameter search procedure described earlier is indicated in **bold**. Fig. 3 shows the learning curves for each algorithm on these data sets in the online setting.

a) GERMAN CREDIT *Data Set:* The goal in this data set is to approve or deny a loan request for an individual, where $\mathbf{x}$ represents the requestor's profile. The cost, $c(\mathbf{x})$, of approving a loan for a requestor who will ultimately default (false positive) is 75% of the requested amount, as requestors typically pay back part of the loan; false negatives have zero cost in this setup.

Table II shows that simply incorporating an example's cost into the standard CW algorithm's update rules can result in overly aggressive updates in the online learning setting, as the performance of CW-costs is actually worse than that of CW; in the batch learning setting, this modification leads to a 3% decrease in cost incurred over the CW algorithm[3]. Introducing an example-level CVaR constraint further reduces the cost incurred by CW-nobuffer (arbitrary margin distribution assumption) over CW-costs by 8% in the online setting, and 1.44% in the batch setting. Finally, introducing a stream-level buffer reduces the cost incurred by CW-CVaR (arbitrary) over CW-nobuffer by 42% in the online setting, and 33% in the batch setting. CW-CVaR (arbitrary) ultimately reduces the cost incurred over CW by 38% in the online setting, and 36% in the batch setting, which outperforms all other algorithms in the online and batch settings; the performance of CW-CVaR under alternate margin distribution assumptions is similar. Table III shows that the CVaR incurred by the algorithms incorporating an example-level CVaR constraint is, naturally, better than that incurred by the other algorithms.

b) KDD CUP 1998 *Data Set:* In this data set, the goal is to identify potential donors. Candidates are sent an invitation to donate, which costs $c = \$0.68$ to assemble and send. The algorithm incurs the additional opportunity cost of a missed



(a) GERMAN CREDIT

(b) KDD CUP 1998

(c) Synthetic

Figure 3. Learning curves for algorithms on real and synthetic data, in the online learning setting (averaged over 100 trials; error bars omitted to reduce clutter).

[2] ϵ, α, and β were allowed to assume values in the set $\{0.01, 0.05, 0.1, 0.2, \ldots, 0.8, 0.9, 0.99\}$.

[3] Note that the test data used in the online and batch settings is not identical, and hence the range of cost incurred may differ.

Table II. COST INCURRED BY ALGORITHMS ON THE GERMAN CREDIT AND KDD CUP 1998 DATA SETS, IN THE ONLINE (O) AND BATCH (B) LEARNING SETTINGS (AVERAGED OVER 100 AND 10 TRIALS, RESPECTIVELY). LOWER COST IS BETTER. **BOLD** INDICATES THE MARGIN DISTRIBUTION ASSUMPTION CHOSEN BY THE HYPER-PARAMETER SEARCH PROCEDURE DESCRIBED IN THE TEXT.

DATA SET	CW	CW-COSTS	CW-CV-ARB-NOBUF	CW-CV-GAUSS	CW-CV-SYMUNI	CW-CV-SYM	CW-CV-ARB
GERMAN (O)	35,002 ± 172.80	40,070 ± 108.80	36,976 ± 126.78	22,132 ± 47.95	21,923 ± 41.39	21,971 ± 44.41	**21,621 ± 44.11**
GERMAN (B)	3,252.20 ± 41.02	3,140.90 ± 29.38	3,095.80 ± 27.07	2,290.50 ± 35.39	1,999.70 ± 28.35	2,154.50 ± 29.97	**2,074.90 ± 34.58**
KDD CUP (O)	$1,620.90 ± 1.54	$1,621.40 ± 1.54	$1,621.20 ± 1.53	$1,532.30 ± 1.97	$1,506.80 ± 1.90	$1,505.20 ± 1.88	**$1,480.30 ± 2.11**
KDD CUP (B)	$214.82 ± 1.83	$215.9 ± 1.77	$216.04 ± 1.77	$202.82 ± 1.77	$174.06 ± 1.39	$173.11 ± 1.66	**$171.22 ± 1.45**

Table III. CVAR COMPUTED AT $\epsilon = 0.05$ ON EACH ALGORITHM'S EMPIRICAL MARGIN DISTRIBUTION ON THE TEST DATA IN THE GERMAN CREDIT AND KDD CUP 1998 DATA SETS, IN THE ONLINE (O) AND BATCH (B) LEARNING SETTINGS. HIGHER CVAR IS BETTER. **BOLD** INDICATES THE MARGIN DISTRIBUTION ASSUMPTION CHOSEN BY THE HYPER-PARAMETER SEARCH PROCEDURE DESCRIBED IN THE TEXT.

DATA SET	CW	CW-COSTS	CW-CV-ARB-NOBUF	CW-CV-GAUSS	CW-CV-SYMUNI	CW-CV-SYM	CW-CV-ARB
GERMAN (O)	-7.65 ± -7.65	-10.65 ± -10.64	-8.30 ± -8.29	-2.82 ± -2.82	-3.20 ± -3.20	-3.28 ± -3.28	**-3.56 ± -3.56**
GERMAN (B)	-0.40 ± -0.40	-0.31 ± -0.31	-0.30 ± -0.29	-0.31 ± -0.31	-0.23 ± -0.22	-0.24 ± -0.24	**-0.27 ± -0.26**
KDD CUP (O)	$-3.47 ± -3.47	$-4.32 ± -4.32	$-3.09 ± -3.09	$-2.3 ± -2.30	$-1.87 ± -1.87	$-1.82 ± -1.81	**$-1.53 ± -1.52**
KDD CUP (B)	$-0.18 ± -0.17	$-0.25 ± -0.24	$-0.16 ± -0.16	$-0.11 ± -0.10	$-0.07 ± -0.07	$-0.08 ± -0.07	**$-0.06 ± -0.06**

donation if a candidate is not sent a mailing when they would have provided a donation (false negative); false positives only incur the cost of sending the mailing.

Tables II and III show similar trends as in the GERMAN CREDIT data set: CW-CVaR reduced the cost incurred over CW by 9% in the online setting, and 20% in the batch setting. This improvement is somewhat less pronounced than in the GERMAN CREDIT data set, and may be due to the small number of donors present in the data set (5%), as well as the small potential upside to classifying a candidate correctly (the maximum donation amount was $200).

B. Synthetic Data Set

We generated a synthetic data set for robust, cost-sensitive binary classification as follows. First, we drew 400 points comprising the positive class from a Gaussian with mean (5,5), and 400 points comprising the negative class from a Gaussian with mean (0,5); both Gaussians had covariance 2**I**. We then drew 200 points from a Gaussian with mean (0,0) and randomly assigned 50% of them to the positive class, and the rest to the negative class: this models unreliability in x_2 in the feature vector (x_1, x_2). To account for this unreliability in CW-robust, we encouraged a weight of 0 for x_2 in the posterior by setting $\nu_2 = -\mu_2$, and $\mathbf{\Upsilon} = \text{diag}(\infty, 1)$. Fig. 3(c) presents the learning curves for CW, CW-CVaR, and CW-robust in the online learning setting, and shows that CW-robust outperforms the other algorithms by incorporating prior knowledge on feature corruption.

VII. CONCLUSION

We presented a new, robust, cost-sensitive, confidence weighted (CW) online learning algorithm that replaced the value at risk (VaR) constraint in the original CW optimization problem with a worst case conditional value at risk (CVaR) constraint. This constraint can be customized by the user to achieve the appropriate level of cost-sensitivity for the task at hand, and led to consistent reductions in cost incurred over the standard CW algorithm and cost-sensitive CW baselines. The new algorithm is also able to incorporate prior knowledge on adversarial feature corruption, which led to further improvements in classification performance. Finally, we showed a connection between the VaR and CVaR constraints used by the CW and CW-CVaR algorithms and robust optimization, which may enable the derivation of new cost-sensitive CW algorithms.

REFERENCES

[1] M. Dredze, K. Crammer, and F. Pereira, "Confidence-weighted linear classification," in *Proceedings of the 25th international conference on Machine learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 264–271.

[2] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Identifying suspicious urls: an application of large-scale online learning," in *ICML*, 2009, p. 86.

[3] B. Zadrozny, J. Langford, and N. Abe, "Cost-sensitive learning by cost-proportionate example weighting," in *Proceedings of the Third IEEE International Conference on Data Mining*, ser. ICDM '03. Washington, DC, USA: IEEE Computer Society, 2003, pp. 435–.

[4] H. Markowitz, "Portfolio Selection," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, Mar. 1952.

[5] E. Even-Dar, M. Kearns, and J. Wortman, "Risk-sensitive online learning," in *Proceedings of the 17th international conference on Algorithmic Learning Theory*, ser. ALT'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 199–213.

[6] R. T. Rockafellar and S. Uryasev, "Optimization of conditional value-at-risk," *Journal of Risk*, vol. 2, pp. 21–41, 2000.

[7] H. Kashima, "Risk-sensitive learning via expected shortfall minimization."

[8] A. Takeda and M. Sugiyama, "Nu-support vector machine as conditional value-at-risk minimization," in *Proceedings of the 25th international conference on Machine learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 1056–1063.

[9] F. Perez-Cruz, J. Weston, D. Hermann, and B. Schoelkopf, "Extension of the nu-svm range for classification," in *Advances in Learning Theory*. IOS Press, 2003.

[10] G. R. Lanckriet, L. E. Ghaoui, C. Bhattacharyya, and M. I. Jordan, "Minimax probability machine," in *Advances in Neural Information Processing Systems 14*. MIT Press, 2001.

[11] B. Li, S. C. H. Hoi, P. Zhao, and V. Gopalkrishnan, "Confidence weighted mean reversion strategy for on-line portfolio selection," *Journal of Machine Learning Research - Proceedings Track*, vol. 15, pp. 434–442, 2011.

[12] K. Crammer, A. Kulesza, and M. Dredze, "Adaptive regularization of weight vectors," in *Advances in Neural Information Processing Systems 22*, 2009, pp. 414–422.

[13] L. El Ghaoui, M. Oks, and F. Oustry, “Worst-case value-at-risk and robust portfolio optimization: A conic programming approach,” *Oper. Res.*, vol. 51, no. 4, pp. 543–556, 2003.

[14] R. T. Rockafellar and S. Uryasev, “Conditional value-at-risk for general loss distributions,” *Journal of Banking & Finance*, vol. 26, no. 7, pp. 1443–1471, Jul. 2002.

[15] Y. Yu, Y. Li, D. Schuurmans, and C. Szepesvári, “A general projection property for distribution families,” in *NIPS*, 2009, pp. 2232–2240.

[16] W. Chen, M. Sim, J. Sun, and C.-P. Teo, “From cvar to uncertainty set: Implications in joint chance-constrained optimization,” *Operations Research*, vol. 58, no. 2, pp. 470–485, 2010.