A Framework of Online Learning with Imbalanced Streaming Data

Yan Yan,[1] **Tianbao Yang,**[2] **Yi Yang,**[1] **Jianhui Chen**[3]
[1]QCIS, University of Technology Sydney, 15 Broadway, Ultimo NSW 2007, Australia
[2]Department of Computer Science, The University of Iowa, Iowa City, IA 52242, USA
[3]Yahoo! Labs, Sunnyvale, CA 94089, USA
yan.yan-3@student.uts.edu.au, tianbao-yang@uiowa.edu, yi.yang@uts.edu.au, jianhuic@gmail.com

Abstract

A challenge for mining large-scale streaming data overlooked by most existing studies on online learning is the skew-distribution of examples over different classes. Many previous works have considered cost-sensitive approaches in an online setting for streaming data, where fixed costs are assigned to different classes, or ad-hoc costs are adapted based on the distribution of data received so far. However, it is not necessary for them to achieve optimal performance in terms of the measures suited for imbalanced data, such as F-measure, area under ROC curve (AUROC), area under precision and recall curve (AUPRC). This work proposes a general framework for online learning with imbalanced streaming data, where examples are coming sequentially and models are updated accordingly on-the-fly. By simultaneously learning multiple classifiers with different cost vectors, the proposed method can be adopted for different target measures for imbalanced data, including F-measure, AUROC and AUPRC. Moreover, we present a rigorous theoretical justification of the proposed framework for the F-measure maximization. Our empirical studies demonstrate the competitive if not better performance of the proposed method compared to previous cost-sensitive and resampling based online learning algorithms and those that are designed for optimizing certain measures.

Introduction

Streaming data are pervasive in many domains, including online social media (Lukasik and Cohn 2016; Akbari et al. 2016), clickbait prediction (Biyani, Tsioutsiouliklis, and Blackmer 2016), ad placement (Liu and Liu 2016), *etc.*. In these scenarios, data are coming sequentially. Mining the streaming data requires the learner to make a prediction instantly after receiving an example and update the model based on the received true label. As the increasing popularity of streaming data, it becomes critical to design effective learning algorithms for mining streaming data and making accurate predictions **on the fly**.

Online learning has emerged to be an important learning paradigm due to its ability to handle streaming data. Different from traditional batch learning, in online learning, data arrive sequentially, and the prediction is made before getting a feedback about the true label. Thus, the online performance of a learner is a critical concern in online learning, since it measures how much the predictions are consistent with the true label.

In most existing studies of online learning, a challenge for mining large-scale streaming data is that examples are usually skew-distributed over different classes. Particularly for binary problems, the number of positive examples is usually significantly smaller than that of negative ones in many applications. Therefore, the zero-one loss and its surrogates commonly used in traditional online learning algorithms are not appropriate for imbalanced data. This issue has been long recognized as cost asymmetry, *i.e.*, the cost for a false negative should be different from that for a false positive. To deal with it, cost-sensitive algorithms, one of the most popular approaches for tackling imbalanced data, have been recently studied in the online setting (Wang, Zhao, and Hoi 2012), which usually assign fixed costs, or ad-hoc costs based on the distribution of data received so far to different classes. However, it would not necessarily achieve superior performance measures including F-measure, area under ROC curve (AUROC), area under precision and recall curve (AUPRC).

Another line of research for learning with the imbalanced streaming data is to directly optimize target measures in an online fashion, which attracts increasing attention recently (Gao et al. 2013; Zhao et al. 2011). However, there are two main limitations. Firstly, measures applied in imbalanced problems, *e.g.*, F-measure, AUROC and AUPRC, are usually not decomposable, which makes it sigfinicantly challenging to directly optimize these measures in the online setting. Moreover, an algorithm designed for optimizing a specific measure (*e.g.*, F-measure) is usually not applicable for optimizing another certain measure (*e.g.*, AUROC).

To address these issues, in this work, we present a unified framework for learning with imbalanced streaming data that is easily adapted to different performance measures. The proposed framework simultaneously learns multiple classifiers with various cost vectors. In particular, at each iteration, the prediction is made by a classifier which is selected randomly according to a sampling distribution, which is updated based on the current performance measures of classifiers, similarly to the well-know exponential weighted average algorithm (Littlestone and Warmuth 1994). The selection of the optimal classifier is adaptive and evolving according to the streaming data. We emphasize that the proposed approach is different from the cross-validation approach, which replies on a separate validation set. Furthermore, the proposed framework enjoys a rigorous theoretical justification for the F-measure maximization. Empirical studies demonstrate that the proposed algorithm is more effective than previous online learning algorithms for imbalanced streaming data.

The remainder of the paper is organized as follows. We firstly review some related work. Secondly we present a framework of online multiple cost-sensitive learning together with some analysis. In the next section, we discuss the application of online learning with the F-measure as the target measure, and present a theoretical analysis of the proposed algorithm for the F-measure maximization. Then we discuss the application of online learning with AUROC and AUPRC as the target measures. In particular, we focus on how to efficiently update AUROC and AUPRC in an online fashion.

Related Work

In traditional online learning, studies revolve around the regret analysis of algorithms for sequential prediction problems (*e.g.*, prediction with expert advice, online classification) (Cesa-Bianchi and Lugosi 2006; Herbster and Warmuth 1998; Littlestone 1988). In these studies, many online algorithms have been developed, *e.g.*, the exponentially weighted average algorithm (Littlestone and Warmuth 1994) and the online gradient descent (Zinkevich 2003). In the last ten years, we observe substantial applications of these algorithms in machine learning and data analytics, *e.g.*, online classification (Gentile 2002; Crammer and Singer 2003).

Learning with cost asymmetry has attracted much attention recently. Most studies cast the problem into cost-sensitive learning that assigns different costs to mistakes of different classes (Elkan 2001; Masnadi-Shirazi and Vasconcelos 2010; Scott 2011). While there exist a long list of literatures on batch learning with cost-sensitivity, few studies were devoted to online learning with cost-sensitivity (Crammer et al. 2006; Wang, Zhao, and Hoi 2012). These studies assume a given cost vector (or matrix) and modify conventional loss functions to incorporate the given cost vector/matrix. The issue with this approach is that the cost vector/matrix is usually unknown when applying to imbalanced data. Recent studies have found that the optimal costs assigned to different classes have an explicit relationship with the optimal performance measure (Puthiya Parambath, Usunier, and Grandvalet 2014). Besides the cost-sensitive approach, some resampling based methods are proposed to deal with imbalanced data. However, most of them focus on batch learning, *e.g.*, (Liang and Cohn 2013), while there are a few works concerning the online setting, *e.g.*, (Wang, Minku, and Yao 2015).

Recently, there emerge some works about online optimization for a particular performance measure, *e.g.*, F-measure, AUROC. For example, (Zhao et al. 2011; Gao et al. 2013) proposed online learning algorithms for AUROC optimization. However, both works focus on the offline performance evaluation. In (Busa-Fekete et al. 2015), the authors proposed an online learning algorithm for F-measure optimization with an automatic threshoding strategy based on the online F-measure. However, they innocently ignored the strategy for updating the model by simply assuming a given algorithm that can learn the posterior probability $\Pr(y|\mathbf{x})$. In (Hu et al. 2015), a method is proposed to directly optimize AUROC, but requires extra resources to store the learned support vectors. The authors in (Kar, Narasimhan, and Jain 2014) proposed an online learning framework for non-decomposable loss functions based on the structural SVM. The drawback of this method is that their online learning algorithm needs to solve a difficult optimization problem at each iteration. As for AUPRC, there still lacks of efforts. Recent studies (Goadrich, Oliphant, and Shavlik 2006; Davis and Goadrich 2006) have found that when dealing with highly skewed datasets, Precision-Recall (PR) curves might give a more informative picture of an algorithm's performance, which gives the measure of AUPRC.

Table 1: Notations (subindex t refers to the t-th round in online learning.)

Notations	Meaning	Notations	Meaning
$\mathbf{x}_t \in \mathbb{R}^d$	feature vector	$\ell(yf(\mathbf{x}))$	loss
$y_t \in \{1, -1\}$	class label	$\hat{y}_t$	$\mathbb{I}(f_t > 0)$
$f_t(\cdot)$	prediction func.	$\bar{y}_t$	$\frac{y_t+1}{2}$
$f_t = f_t(\mathbf{x}_t)$	prediction	$\sigma(f)$	$\frac{1}{1+\exp(-f)}$
$\mathbb{I}(b)$	indicator func.	$\mathbf{p}_t \in \mathbb{R}^K$	sampl. probs.
M_t: performance measure based on $\{\hat{y}_\tau, y_\tau, \tau = 1, \ldots, t-1\}$			

Finally, we note that the proposed algorithm is different from online Bayesian learning that maintains and updates the posterior distribution of model parameters (Dredze, Crammer, and Pereira 2008), and is also different from the online ensemble algorithm in (Vovk 1990) that aggregates all classifiers for prediction. The synthesis of online gradient descent for updating individual classifiers and the exponential weighted average algorithm for updating probabilities is similar to the work of online kernel selection (Yang et al. 2012). However, the two work have different focuses. In particular, their goal is to select the best kernel classifier among multiple kernel classifiers for optimizing traditional measures while our goal is to select the best cost-sensitive classifier among multiple cost-sensitive classifiers for optimizing a target measure suited for imbalanced data. Therefore, their analysis can not be borrowed for our purpose.

Online Multiple Cost-Sensitive Learning

We first present some notations. Let $\mathbf{x}_t \in \mathbb{R}^d$ denote the feature vector of the example received at the t-th iteration, and $y_t \in \{1, -1\}$ denote its true class label. We denote by $f_t(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$ a prediction function at the t-th iteration and by $f_t = f_t(\mathbf{x}_t)$ the prediction on the t-th example. Let $\mathbb{I}(b)$ denote an indicator function, where $\mathbb{I}(b) = 1$ if b is true and 0 otherwise. Commonly used notations in this paper are summarized in Table 1.

In traditional online learning, the performance of $f_t(\cdot)$ on the example $\mathbf{x}_t$ is usually measured by a loss function

$\ell(y_t f_t(\mathbf{x}_t))$, *e.g.*, hinge loss $\ell(z) = \max(0, 1-z)$ and logistic loss $\ell(z) = \log(1+\exp(-z))$, which are considered to be a surrogate loss of 0-1 error $\mathbb{I}(sign(f_t(\mathbf{x}_t)) \neq y_t)$. Previous studies cast the problem into learning a sequence of classifiers $f_1(\cdot), \ldots, f_T(\cdot)$ such that the regret defined below is minimized, $R_T = \sum_{t=1}^T \ell(y_t f_t(\mathbf{x}_t)) - \min_f \sum_{t=1}^T \ell(y_t f(\mathbf{x}_t))$. Many online learning algorithms have been proposed to minimize the regret such as online gradient descent (Zinkevich 2003)). However, a critique over the standard surrogate loss functions is that they ignore the cost asymmetry between the majority class and the minority one. To resolve this issue, cost-sensitive loss functions have been proposed, which give different costs to different classes: $\ell_c(f(\mathbf{x}), y) = c_+\mathbb{I}(y=1)\ell(f(\mathbf{x})) + c_-\mathbb{I}(y=-1)\ell(-f(\mathbf{x}))$, where $\mathbf{c} = (c_+, c_-)$ is the cost vector that controls the balance between the two loss terms. How to decide the value of c_+ and c_- remains an issue. Previous works use ad-hoc approaches to set up these parameters (Wang, Zhao, and Hoi 2012). However, there is no guarantee that these ad-hoc approaches use appropriate values for c_+ and c_-. In addition, if c_+ and c_- are changing during the training, it is difficult to analyze the performance of the learned classifier. Another commonly used practice in batch learning is by a cross-validation approach that tunes the values of c_+ and c_- based on the offline performance on a separate validation set. Nevertheless, in online learning a separate validation set is usually not available and even if it is available there is no guarantee that the distribution of the examples in the validation set is the same as the received examples in online learning.

To address these issues, we propose an online learning framework of multiple cost-sensitive learning. The motivation is that if multiple classifiers with a number of $\mathbf{c}$ are learned simultaneously, there must exist one setting that is most appropriate to the data. Without loss of generality, we assume $c_+ + c_- = 1$ and as a result one parameter $c_+ \in (0,1)$ is needed to be set. To construct the pool of multiple values of c_+, we discretize $(0,1)$ into K evenly distributed values $\theta_1, \ldots, \theta_K$, *i.e.*, $\theta_j = j/(K+1)$. With the value of $c_+ = 1 - \theta_j/2$, the corresponding cost sensitive loss is denoted by

$$\begin{aligned}\ell_c^j(f(\mathbf{x}), y) =& (1-\theta_j/2)\mathbb{I}(y=1)\ell(f(\mathbf{x})) \\ &+ (\theta_j/2)\mathbb{I}(y=-1)\ell(-f(\mathbf{x}))\end{aligned} \tag{1}$$

The reason that we divide θ_j by 2 will be clear when we present the theoretical justification. Then we learn K sequences of classifiers $f_t^1(\cdot), f_t^2(\cdot), \ldots, f_t^K(\cdot)$ simultaneously in online learning, with each sequence of $f_t^j(\cdot), t = 1, \ldots, T$ to minimize the associated regret $R_T^j = \sum_{t=1}^T \ell_c^j(f_t^j(\mathbf{x}_t), y_t) - \min_f \sum_{t=1}^T \ell_c^j(f(\mathbf{x}_t), y_t)$.

A remaining issue is how to choose a classifier from K candidates to predict $\mathbf{x}_t$ at the t-th iteration. Based on our motivation, a greedy approach is to track the "performance" of K classifiers and select the best performer on historical examples. However, it may lead to overfitting problems. We thus propose a theoretically sound randomized method that selects a classifier for prediction according to a distribution $\mathbf{p}_t = (p_t^1, \ldots, p_t^K)^\top$ such that $\sum_j p_t^j = 1$ and $p_j^t \geq 0$. To compute the sampling probabilities, we use the following formula

$$p_t^j = \frac{\exp(\gamma M_t^j)}{\sum_{j=1}^K \exp(\gamma M_t^j)}, \quad j = 1, \ldots, K, \tag{2}$$

where $\gamma > 0$ is a learning rate hyper-parameter, and M_t^j is some favorite performance measure (the higher the better, *e.g.*, F-measure, AUROC, AUPRC, *etc.*) on historical examples $(\mathbf{x}_\tau, y_\tau), \tau = 1, \ldots, t-1$ using the predictions $f_1^j, \ldots, f_{t-1}^j$ of the j-th sequence of classifiers. From the Equation (2), classifier with higher performance will have a higher probability to be selected for making the prediction. Note that when $\gamma \to \infty$, the above approach reduces to the greedy approach. We would like to emphasize that the sampling probabilities defined above are similar to that in exponentially weighted average algorithm (Littlestone and Warmuth 1994) for selecting the best expert advice but with a key difference. In the learning with expert advice problem, the sampling probabilities are computed based on the cumulative loss $\sum_{\tau=1}^{t-1} \ell_t^j$ of different experts indexed by j, while our sampling probabilities are computed based on interesting performance measure that is suited for imbalanced data.

Algorithm 1 A Framework of Online Multiple Cost-sensitive Learning

```
1:  Input: the number of classifiers $K$
2:  Initialize $\mathbf{p}_1 = (1/K, \ldots, 1/K)$, $f_1^j(\mathbf{x}) = 0, j = 1, \ldots, K$
3:  for $t = 1, \ldots, T$ do
4:      Receive an example $\mathbf{x}_t$
5:      Sampling a classifier $f_t^j$ by choosing $j_t$ according to $\Pr(j) = p_t^j$
6:      Compute a predicted label $\tilde{y}_t = sign(f_{j_t}^t(\mathbf{x}_t))$
7:      Receive the true label $y_t$
8:      for $j = 1, \ldots, K$ do
9:          Update the classifier $f_{t+1}^j(\cdot) = \mathcal{A}(f_t^j(\cdot), \mathbf{x}_t, y_t)$
10:         Update the performance $M_{t+1}^j = \mathcal{M}(y_{1:t}, f_{1:t}^j)$
11:     end for
12:     Update the sampling probabilities $\mathbf{p}_{t+1}$ according to (2)
13: end for
```

Now we can summarize our online multiple cost-sensitive learning in Algorithm 1. In the remainder of this section, we discuss how to update the classifier in step 9 and present a theoretical analysis of the proposed framework of online multiple cost-sensitive learning (OMCSL). In the next two sections, we discuss the step 10 that updates the performance for different measures.

For updating the classifier, we can use any online learning algorithms as long as they are designed to minimize the regret, such as online gradient descent (OGD) (Zinkevich 2003), online dual averaging (Xiao 2010), follow the regularized leader (Kalai and Vempala 2005), and some specialized algorithms for online classification including online passive aggressive learning (Crammer et al. 2006), perceptron (Shalev-Shwartz and Singer 2005), *etc.*. Due to the popularity and simplicity of OGD, we present the online gradient descent update. For the easy presentation, we here consider $f_t^j(\cdot)$ as a linear function, namely $f_t^j(\mathbf{x}) = \mathbf{x}^\top \mathbf{w}_t^j$. We thus update $\mathbf{w}_t^j$ by:

$$\mathbf{w}_{t+1}^j = \mathbf{w}_t^j - \eta_t \nabla_{\mathbf{w}} \ell_c^j(\mathbf{x}_t^\top \mathbf{w}_t^j, y_t), j = 1, \ldots, K. \tag{3}$$

where η_t is a step size hyper-parameter, which can be set to a small value or to be decreasing depending on the property of the loss function (Zinkevich 2003). It is worth noting that (i) the loss function could include a regularizer on $\mathbf{w}$, *e.g.*, $\frac{\lambda}{2}\|\mathbf{w}\|_2^2$; (ii) a bias term can be incorporated by adding an extra constant feature to $\mathbf{x}$. The proposition below provides the regret guarantee for the j-th sequence of classifiers. For ease of presentation, we specialize to the linear function. One can easily generalize it to a non-linear function from a RKHS.

Proposition 1 *(Theorem 3.1 (Hazan 2015)) Let the linear prediction function $f_t^j(\mathbf{x}) = \mathbf{x}^\top \mathbf{w}_t^j$ be updated based on (3) and $\mathbf{w}_*^j$ be the optimal prediction function that minimizes the cumulative cost-sensitive loss $\sum_{t=1}^T \ell_c^j(\mathbf{w}^\top \mathbf{x}_t, y_t)$. Assume that $\left\|\nabla_{\mathbf{w}} \ell_c^j(y\mathbf{x}^\top \mathbf{w})\right\|_2 \leq G$ and $\left\|\mathbf{w}_*^j\right\|_2 \leq D$. By setting $\eta_t = \frac{D}{G\sqrt{t}}$, then we have*

$$R_T^j = \sum_{t=1}^T \ell_c^j(f_t^j(\mathbf{x}_t), y_t) - \sum_{t=1}^T \ell_c^j(\mathbf{x}_t^\top \mathbf{w}_*^j, y_t) \leq 3GD\sqrt{T},$$

which implies that the averaged regret converges to zero at a rate of $1/\sqrt{T}$, i.e, $\frac{R_T^j}{T} \leq \frac{3GD}{\sqrt{T}}$.

Next, we analyze the updating rule of sampling probabilities in (2). We first present a proposition below and then provide an explanation of it.

Proposition 2 *Let $\mathbf{M}_t = (M_t^1, \ldots, M_t^K)^\top$ and $\mathbf{p}_t$ updated according to (2). Then there exists a $\gamma > 0$ such that $\mathbf{p}_T^\top \mathbf{M}_T \geq \max_{1\leq j\leq K} M_T^j - (V_T + \sqrt{V_T \log K})$, where $V_T = 2\sum_{t=1}^T \|\mathbf{M}_t - \mathbf{M}_{t-1}\|_\infty$ is the scaled sum of consecutive variation of the performance measure.*

From the proposition above, we can see that when the variation of the performance measure is small, the expected performance of the selected classifier (the L.H.S of the inequality) is close to the best performance measure (the R.H.S). We emphasize that the lower bound of $\mathbf{p}_T^\top \mathbf{M}_T$ in Proposition 2 is by no means tight. It only explains to some degree why the employed sampling probabilities make sense. Bounding the online performance of an non-decomposable measure (*e.g.*, F-measure, AUROC and AUPRC) is still very challenging. In next section, we provide a theoretical analysis of the proposed framework for the F-measure optimization.

OMCSL for F-measure

In this section, we first present how to update the online F-measure, and then show that OMCSL has a solid theoretical foundation for F-measure maximization, in which the best classifier among the K classifiers will eventually yield a close-to-optimal F-measure provided that K is sufficiently large.

Given a sequence of labels $y_1, \ldots, y_t$ and a sequence of predictions $f_1, \ldots, f_t$, we can calculate the F-measure by $F_{t+1} = \frac{2\sum_{\tau=1}^t \bar{y}_\tau \hat{y}_\tau}{\sum_{\tau=1}^t \bar{y}_\tau + \sum_{\tau=1}^t \hat{y}_\tau}$ where $\bar{y}_t = (y_t+1)/2 \in \{1, 0\}$ and $\hat{y}_t = \mathbb{I}(f_t > 0)$. However, directly calculating the online F-measure by going through all examples is expensive, which requires to store all predictions $f_t(\mathbf{x}_t)$ and y_t. Indeed, the online F-measure can be calculated incrementally. To this end, we let $a_t = \sum_{\tau=1}^t \bar{y}_\tau \hat{y}_\tau$ and $c_t = \sum_{\tau=1}^t \bar{y}_\tau + \sum_{\tau=1}^t \hat{y}_\tau$. Then we can calculate $F_{t+1} = \frac{2a_t}{c_t}$ and update a_t and c_t incrementally by

$$a_{t+1} = \begin{cases} a_t + 1, \text{ if } y_{t+1} = 1 \text{ and } f_{t+1} > 0, \\ a_t, \text{ otherwise;} \end{cases}$$

$$c_{t+1} = \begin{cases} c_t + 2, \text{ if } y_{t+1} = 1 \text{ and } f_{t+1} > 0, \\ c_t + 1, \text{ if } y_{t+1} = 1 \text{ or } f_{t+1} > 0, \\ c_t, \text{ if } y_{t+1} = -1 \text{ and } f_{t+1} \leq 0. \end{cases} \tag{4}$$

A Theoretical Justification

We show that when K is sufficiently large, there exists a sequence of classifiers among the K sequences that will eventually converge to a classifier that has a close-to-optimal F-measure. To this end, we assume the data is i.i.d. The analysis is built on several previous works on the F-measure maximization (Puthiya Parambath, Usunier, and Grandvalet 2014) and the theory of consistency for cost-sensitive surrogate loss minimization (Scott 2012). To present the results, we first give some notations. Let $h(\mathbf{x}) \in \mathcal{H} : \mathbb{R}^d \to \{1, -1\}$ denote a classifier and $\mathbf{e}(h) = (e_1(h), e_2(h))^\top$ denote the false negative (FN) error and false positive (FP) error of $h(\mathbf{x})$, respectively, *i.e.*, $e_1(h) = \Pr(y = 1, h(\mathbf{x}) = -1)$, $e_2(h) = \Pr(y = -1, h(\mathbf{x}) = 1)$ where $\Pr(\cdot)$ denotes the probability over $(\mathbf{x}, y)$. When it is clear from the context, we write $\mathbf{e} = \mathbf{e}(h)$ for short. Let P_1 denote the marginal probability of the positive class, *i.e.*, $P_1 = \Pr(y = 1)$. Then the F-measure of $h(\cdot)$ on the population level can be computed by (Puthiya Parambath, Usunier, and Grandvalet 2014) $F(h) \triangleq F(\mathbf{e}) = \frac{2(P_1 - e_1)}{2P_1 - e_1 + e_2}$. Let $\mathbf{c}(\tau) = (1 - \frac{\tau}{2}, \frac{\tau}{2})^\top$. The following proposition exhibits that maximizing F-measure is equivalent to minimizing a cost-sensitive error.

Proposition 3 *(Proposition 4 (Puthiya Parambath, Usunier, and Grandvalet 2014)) Let $F_* = \max_{\mathbf{e}} F(\mathbf{e})$. Then we have $\mathbf{e}_* = \arg\min_{\mathbf{e}} \mathbf{c}(F_*)^\top \mathbf{e} \Leftrightarrow F(\mathbf{e}_*) = F_*$.*

The above proposition indicates that one can optimize the following cost-sensitive error

$$\mathbf{c}(F_*)^\top \mathbf{e} = \left(1 - \frac{F_*}{2}\right) e_1 + \frac{F_*}{2} e_2, \tag{5}$$

to obtain an optimal classifier $h^*(\mathbf{x})$, which will give the optimal F-measure, *i.e.*, $F(h^*) = F_*$. However, the cost-sensitive error in (5) requires knowing the exact value of the optimal F-measure. To address this issue, we discretize $(0, 1)$ to have a set of evenly distributed values $\{\theta_1, \ldots, \theta_K\}$ such that $\theta_{j+1} - \theta_j = \epsilon_0/2$, which serve as the candidate values of F_*. Then we can solve for a series of K classifiers to minimize the cost-senstive error

$$h_j^* = \arg\min_{h\in\mathcal{H}} \left(1 - \frac{\theta_j}{2}\right) e_1 + \frac{\theta_j}{2} e_2 = \mathbf{c}(\theta_j)^\top \mathbf{e}, j = 1, \ldots, K. \tag{6}$$

This explains our choice of $\theta_j/2$ in (1). The following proposition shows that there exists one classifier among $\{h_j^*, \cdots, h_K^*\}$ that can achieve a close-to-optimal F-measure as long as ϵ_0 is small enough.

Proposition 4 *Let $\{\theta_1, \ldots, \theta_K\}$ be a set of values evenly distributed in $(0,1)$ such that $\theta_{j+1} - \theta_j = \epsilon_0/2$. Then there exists $h_j^* \in \{h_j^*, \cdots, h_K^*\}$ such that $F(h_j^*) \geq F_* - \frac{2\epsilon_0 B}{P_1}$, where $B = \max_{\mathbf{e}} \|\mathbf{e}\|_2$.*

Remark: The above proposition also implies an interesting result that the smaller P_1 (*i.e.*, more imbalanced of the data), the larger gap between $F(h_j^*)$ and F_* (*i.e.*, more difficult to optimize the F-measure).

Proposition 4 only provides the guarantee on the optimal classifiers $\{h_j^*, \cdots, h_K^*\}$. In practice, one cannot obtain these optimal classifiers because the distribution of the data is unknown. The following proposition shows that as long as the obtained classifiers achieve a cost-sensitive error close to the optimal classifiers, a similar guarantee to that in Proposition 4 holds.

Proposition 5 *Let $\{\theta_1, \ldots, \theta_K\}$ be a set of values evenly distributed in $(0,1)$ such that $\theta_{j+1} - \theta_j = \epsilon_0/2$. Let $\{\hat{h}_1, \ldots, \hat{h}_K\}$ be a set of classifiers that minimize the cost-sensitive errors in (6) to a certain degree such that $\mathbf{c}(\theta_j)^\top \mathbf{e}(\hat{h}_j) \leq \mathbf{c}(\theta_j)^\top \mathbf{e}(h_j^*) + \epsilon_1$. Then there exists $\hat{h}_j$ such that $F(\hat{h}_j^*) \geq F_* - \frac{(2\epsilon_0 B + \epsilon_1)}{P_1}$, where $B = \max_{\mathbf{e}} \|\mathbf{e}\|_2$.*

Remark: The result in Proposition 4 is a special case of Proposition 5 when $\epsilon_1 = 0$. Proposition 5 is a corollary of Proposition 5 in (Puthiya Parambath, Usunier, and Grandvalet 2014).

Finally, we are ready to present the theoretical guarantee on the presented OMCSL algorithm for the F-measure maximization.

Theorem 1 *Let $\{\theta_1, \ldots, \theta_K\}$ be a set of values evenly distributed in $(0,1)$ such that $\theta_{j+1} - \theta_j = \epsilon_0/2$, $\mathbf{w}_t^j, t = 1, \ldots, T$ be a sequence updated according to (3) based on the j-th cost-sensitive loss in (1) such that $\|\mathbf{w}_t^j\|_2 \leq D$, and $\widehat{\mathbf{w}}_T^j = \sum_{t=1}^T \mathbf{w}_t^j / T$. Assume $(\mathbf{x}_t, y_t), t = 1, \ldots, T$ are i.i.d. samples such that $\|\mathbf{x}_t\|_2 \leq R$ and the loss function $\ell(z) = \max(0, 1-z)$ is the hinge loss. There exists a $j \in \{1, \ldots, K\}$ with a probability $1 - \delta$ such that*

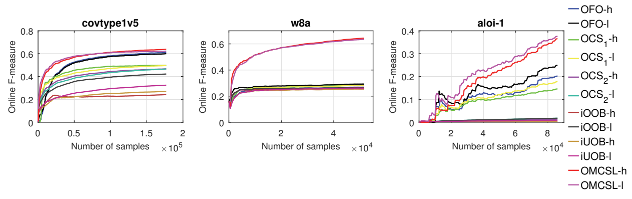
$$F(\hat{h}_T^j) \geq F_* - \frac{2\epsilon_0 B + 3RD(1 + \ln(2/\delta))/\sqrt{T}}{P_1}$$

where $\hat{h}_T^j(\mathbf{x}) = sign(\mathbf{x}^\top \widehat{\mathbf{w}}_T^j)$.

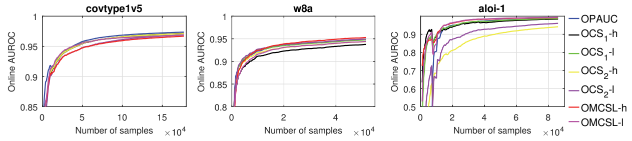
Remark: The theorem implies that when $T \to \infty$, there exists a classifier $\hat{h}_T^j = sign(\mathbf{x}^\top \widehat{\mathbf{w}}_T^j)$ achieves a close-to-optimal F-measure as long as ϵ_0 is small enough. The proof is presented in the supplement.

OMCSL for AUROC and AUPRC

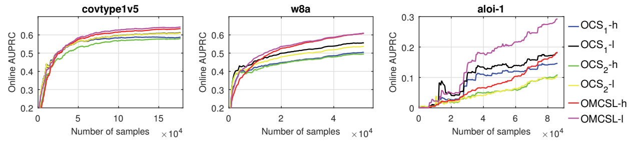
In this section, we briefly present how to update AUROC and AUPRC in an online fashion. The challenge of updating AUROC and AUPRC in the online setting lies at that we need to compare the present example to historically received examples in terms of predictions. A naive way to achieve this is to store the labels and predictions of all classifiers for historically received examples. However, this would increase the memory requirements, which is usually not allowed in online learning. To avoid storing the labels and predictions of all examples, we introduce two hash tables L_t^+ and L_-^t with a fixed length of m that partitions $(0,1)$ into m ranges $(0, 1/m), (1/m, 2/m), \ldots, ((m-1)/m, 1)$. For $i \in \{1, \ldots, m\}$, $L_t^+[i]$ stores the number of positive examples before the t-th iteration (including the t-th iteration) whose predictions f are such that $\sigma(f) \in [(i-1)/m, i/m)$ [1], and $L_t^-[i]$ stores the number of negative examples before the t-th iteration (including t-th iteration) whose predictions f are such that $\sigma(f) \in [(i-1)/m, i/m)$.



(a) Average online F-measure



(b) Average online AUROC



(c) Average online AUPRC

Figure 1: Online performance.

Given L_t^+ and L_t^-, we can show that AUROC$_{t+1}$ can be updated approximately using the two hash tables. In particular, if $y_{t+1} = 1$, we have AUROC$_{t+1} = \frac{N_t^+}{N_t^+ + 1}$AUROC$_t$ + $\frac{1}{(N_t^+ + 1)N_t^-}\left(\sum_{j=1}^{i} L_t^-[j] + L_t^-[i+1]/2\right)$, where i is the largest index such that $i/m \leq \sigma(f_{t+1})$, and if $y_{t+1} = -1$, we update it by AUROC$_{t+1} = \frac{N_t^-}{N_t^- + 1}$AUROC$_t$ + $\frac{1}{N_t^+(N_t^- + 1)}\left(\sum_{j=i+1}^{m-1} L_t^+[j] + L_t^+[i]/2\right)$, where i is the smallest index such that $i/m \geq \sigma(f_{t+1})$.

Similarly, by using L_t^+ and L_t^-, we derive the online update of AUPRC$_t$ as below: AUPRC$_{t+1} = \frac{1}{2}\sum_{i=0}^{m-1}(\mathrm{R}(i) - \mathrm{R}(i+1))(\mathrm{P}(i) + \mathrm{P}(i+1))$. where $\mathrm{R}(i) = \frac{\sum_{j=i+1}^{m} L_t^+[j]}{N_t^+}$, and $\mathrm{P}(i) = \frac{\sum_{j=i+1}^{m} L_t^+[j]}{\sum_{j=i+1}^{m} L_t^+[j] + \sum_{j=i+1}^{m} L_t^-[j]}$. The overall time complexity of computing AUROC$_{t+1}$ and AUPRC$_{t+1}$ is $O(m)$. Detailed development of the online update of AUROC and AUPRC can be found in Appendix C.

Experiments

In this section, we evaluate OMCSL for optimizing three measures, F-measure, AUROC and AUPRC, and compare

[1] $\sigma(f)$ is the sigmoid function defined in Table 1.

Table 2: Data statistics.

Datasets	#Examples	#Features	#Pos:#Neg
covtype1v5	211,840	54	1:22.3
w8a	64,700	300	1:32.5
aloi-1	108,000	128	1:999

with competing online learning algorithms on three public imbalanced datasets. Table 2 lists the statistics of used three datasets. To construct imbalanced data from multiclass datasets `covtype`, we sample instances of the fifth class as positive and instances of the first class as negative, denoted by `covtype1v5`. Similarly, for `aloi`, we sample instances of the first class as positive, and the rest as negative, denoted by `aloi-1`. For each dataset, we randomly sample 4/5 instances as the training set and the rest 1/5 as the testing set. We repeat the experiment on 25 various random splits and report the average results.

We compare the proposed OMCSL method with several state of the art online learning algorithms, namely OCS_1, OCS_2 (Wang, Zhao, and Hoi 2012), OFO (Busa-Fekete et al. 2015), OPAUC (Gao et al. 2013), and iOOB, iUOB (Wang, Minku, and Yao 2015). Among them, OFO and OPAUC directly optimize the target measures (*i.e.*, F-measure and AUROC, respectively), OCS_1 and OCS_2 are both cost-sensitive online methods, iOOB and iUOB are resampling based ensemble methods (oversampling and undersampling). Since the latter two algorithms apply voting to predict a new instance, rather than decision values, we only compute F-measure for them. To examine the performance of using different loss functions, we investigate both the hinge loss and the logistic loss in the experiment and denote these two loss functions by suffixing "-h" and "-l" to the corresponding methods respectively. Note that OPAUC is designed only for square loss, thus we only report one result for OPAUC. The details of hyperparameters of these methods can be found in Appendix D.

Results

We evaluate and compare both online performance on training data and testing performance on testing data. Note that the testing performance is to evaluate the returned models on the testing data in batch, which demonstrates the generalization ability of different online learning algorithms. Table 3 lists the prediction performance on testing data of various algorithms. Fig. 1(a), Fig. 1(b) and Fig. 1(c) demonstrate the averaged online performance (*i.e.*, F-measure, AUROC and AUPRC) of various algorithms on three datasets over 25 trials. As can be observed from both online performance and testing performance, OMCSL achieves better performance than cost-sensitive online algorithms and resampling based online algorithms. The three figures also exhibit a clear trend that when the ratio of positive examples to the negative examples increases, the advantage of OMCSL becomes more striking. Compared to the methods that directly optimize target measure, *i.e.*, OFO and OPAUC, OMCSL achieves competitive if not better performance.

An important reason that OMCSL achieves satisfactory performance is the capability to select a close-to-optimal cost vector $[c_+, c_-]$, which is also exhibited by our theoretical analysis for F-measure optimization. To investigate this property, we perform online cost-sensitive learning with the same costs used in OMCSL, *i.e.*, $c_+ = \{0.55, 0.60, 0.65, ..., 0.95\}$, respectively to find the best cost according to the overall online performance, which we denote by c_+^*. To compare the selected best cost (corresponding to the largest selection probability) by the proposed OMCSL, we average c_+ selected by OMCSL in the last 5,000 iterations as an estimate of the best cost, which we denote by $\hat{c}_+$. Then we compute the absolute error between c_+^* and $\hat{c}_+$, *i.e.*, $err = |c_+^* - \hat{c}_+|$, and report the average error over 25 trials in Table 4. Our observation is that $\hat{c}_+$ predicted by OMCSL is often close to c_+^* given that the step length for search of c_+ is set to 0.05. This property is particularly crucial in the online scenario due to the requirement of going through the training data only once. The proposed OMCSL provides an accurate estimation of the optimal cost.

Conclusion

This work presents a unified online learning framework for imbalanced data. The proposed algorithm simultaneously trains multiple classifiers with various costs, and predicts by randomly selecting a classifier based on a distribution determined by online performance of individual learners. A rigorous theoretical justification for the F-measure maximization is provided. Empirical studies show the superior performance of OMCSL and its capability to select satisfactory costs.

Acknowledgement

Y. Yan and Y. Yang are in part supported by the Data to Decisions Cooperative Research Centre www.d2dcrc.com.au. T. Yang is partially supported by National Science Foundation (IIS-1463988, IIS-1545995). Most work of Y. Yan was done when he was visiting the University of Iowa.

References

Akbari, M.; Huc, X.; Liqianga, N.; and Chua, T.-S. 2016. From tweets to wellness: Wellness event detection from twitter streams. In *AAAI*.

Biyani, P.; Tsioutsiouliklis, K.; and Blackmer, J. 2016. 8 amazing secrets for getting more clicks: Detecting clickbaits in news streams using article informality.

Busa-Fekete, R.; Szörényi, B.; Dembczynski, K.; and Hüllermeier, E. 2015. Online f-measure optimization. In *NIPS*, 595–603.

Cesa-Bianchi, N., and Lugosi, G. 2006. *Prediction, Learning, and Games*. Cambridge University Press.

Crammer, K., and Singer, Y. 2003. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research* 3:951–991.

Crammer, K.; Dekel, O.; Keshet, J.; Shalev-Shwartz, S.; and Singer, Y. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research* 7.

Davis, J., and Goadrich, M. 2006. The relationship between precision-recall and roc curves. In *ICML*, 233–240.

Dredze, M.; Crammer, K.; and Pereira, F. 2008. Confidence-weighted linear classification. In *ICML*.

Table 3: Average prediction performance on the testing set over 25 trials.

Methods	covtype1v5			w8a			aloi-1		
	Fmeasure	AUROC	AUPRC	Fmeasure	AUROC	AUPRC	Fmeasure	AUROC	AUPRC
OPAUC	–	**0.9813**	–	–	**0.9602**	–	–	**0.9993**	–
OFO-h	**0.7071**	–	–	0.6616	–	–	0.2596	–	–
OCS$_1$-h	0.5204	0.5000	0.4999	0.4948	0.4761	0.4726	0.3148	0.4285	0.3148
OCS$_2$-h	0.5035	0.5035	0.4820	0.4478	0.4478	0.4478	0.1062	0.1204	0.0311
iOOB-h	0.1180	–	–	0.0837	–	–	0.0021	–	–
iUOB-h	0.1174	–	–	0.0839	–	–	0.0021	–	–
OMCSL-h	0.6449	**0.9809**	**0.7042**	**0.7147**	**0.9598**	**0.7087**	**0.4560**	**0.9996**	**0.7732**
OFO-l	**0.6600**	–	–	0.6325	–	–	0.1407	–	–
OCS$_1$-l	0.5230	0.5627	0.5230	0.5156	0.5156	0.6381	0.4473	0.4966	0.6176
OCS$_2$-l	0.5044	0.5044	0.5044	0.4405	0.4511	0.6241	0.1429	0.0237	0.4760
iOOB-l	0.1356	–	–	0.0907	–	–	0.0038	–	–
iUOB-l	0.1256	–	–	0.0903	–	–	0.0026	–	–
OMCSL-l	0.6597	**0.9823**	**0.7187**	**0.6891**	**0.9551**	**0.7086**	**0.5197**	**0.9998**	**0.8208**

† Suffixes "-h" and "-l" stand for the algorithms with hinge loss and logistic loss respectively. The top results are in bold.

* For OFO and OPAUC which directly optimize a specific measure, we omit their results in other measures indicated by "–". iOOB and iUOB are resampling based ensemble algorithms which predict a new instance by voting, rather than decision values. Therefore, AUROC and AUPRC are unavailable, indicated by "–".

Table 4: Average absolute error between $\hat{c}_+$ predicted by OMCSL and c_+^*.

Methods	covtype1v5			w8a			aloi-1		
	Fmeasure	AUROC	AUPRC	Fmeasure	AUROC	AUPRC	Fmeasure	AUROC	AUPRC
OMCSL-h	0.010	0.004	0.052	0.059	0.052	0.137	0.084	0.016	0.154
OMCSL-l	0.005	0	0	0.078	0.023	0.127	0.144	0.003	0.165

Elkan, C. 2001. The foundations of cost-sensitive learning. In *IJCAI*.

Gao, W.; Jin, R.; Zhu, S.; and Zhou, Z. 2013. One-pass AUC optimization. In *ICML*, 906–914.

Gentile, C. 2002. A new approximate maximal margin classification algorithm. *Journal of Machine Learning Research* 2:213–242.

Goadrich, M.; Oliphant, L.; and Shavlik, J. W. 2006. Gleaner: Creating ensembles of first-order clauses to improve recall-precision curves. *Machine Learning* 64(1-3):231–261.

Hazan, E. 2015. *Introduction to Online Convex Optimization*. now Publishers Inc.

Herbster, M., and Warmuth, M. K. 1998. Tracking the best expert. *Machine Learning* 32(2):151–178.

Hu, J.; Yang, H.; King, I.; Lyu, M. R.; and So, A. M.-C. 2015. Kernelized online imbalanced learning with fixed budgets. In *AAAI*, 2666–2672.

Kalai, A., and Vempala, S. 2005. Efficient algorithms for online decision problems. *J. Comput. Syst. Sci.* 291–307.

Kar, P.; Narasimhan, H.; and Jain, P. 2014. Online and stochastic gradient methods for non-decomposable loss functions. In *NIPS*, 694–702.

Liang, G., and Cohn, A. G. 2013. An effective approach for imbalanced classification: Unevenly balanced bagging. In *AAAI*, 1633–1634. AAAI Press.

Littlestone, N., and Warmuth, M. K. 1994. The weighted majority algorithm. *Information and Computation* 212–261.

Littlestone, N. 1988. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning* 2:285–318.

Liu, Y., and Liu, M. 2016. Finding ones best crowd: Online learning by exploiting source similarity.

Lukasik, M., and Cohn, T. 2016. Convolution kernels for discriminative learning from streaming text. In *AAAI*.

Masnadi-Shirazi, H., and Vasconcelos, N. 2010. Risk minimization, probability elicitation, and cost-sensitive svms. In *ICML*.

Puthiya Parambath, S.; Usunier, N.; and Grandvalet, Y. 2014. Optimizing f-measures by cost-sensitive classification. In *NIPS*.

Scott, C. 2011. Surrogate losses and regret bounds for cost-sensitive classification with example-dependent costs. In *ICML*.

Scott, C. 2012. Calibrated asymmetric surrogate losses. *Electronic Journal of Statistics* 6:958–992.

Shalev-Shwartz, S., and Singer, Y. 2005. A new perspective on an old perceptron algorithm. In *COLT*.

Vovk, V. 1990. Aggregating algorithms. In *COLT*.

Wang, S.; Minku, L. L.; and Yao, X. 2015. Resampling-based ensemble methods for online class imbalance learning. *IEEE Transactions on Knowledge and Data Engineering* 27(5):1356–1368.

Wang, J.; Zhao, P.; and Hoi, S. C. H. 2012. Cost-sensitive online classification. In *ICDM*, 1140–1145.

Xiao, L. 2010. Dual averaging methods for regularized stochastic learning and online optimization. *J. Mach. Learn. Res.* 11:2543–2596.

Yang, T.; Mahdavi, M.; Jin, R.; Yi, J.; and Hoi, S. C. H. 2012. Online kernel selection: Algorithms and evaluations. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada.*

Zhao, P.; Hoi, S. C. H.; Jin, R.; and Yang, T. 2011. Online auc maximization. In *ICML*, 233–240.

Zinkevich, M. 2003. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, 928–936.