A Framework of Online Learning with Imbalanced Streaming Data

Team 19

- Abhishek Garg
- Aditya Jain
- Prashant Tyagi
- Shashank Kumar Singh

Introduction

Streaming Data is data that is continuously generated by different sources. Such data should be processed incrementally using Stream Processing techniques without having access to all of the data.

Online Learning : Online learning is used to predict the label of the streaming data. Later when the actual label is available, that is used to update the prediction model.

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.

Accuracy of the online learning model can be determined based on how predicted labels match with the actual labels.

Introduction(conti..)

- Online Learning finds application in many places where predictions are required on the go.
- Examples are spam filtering, stock market,etc.

Challenges in online learning

- Examples are skew-distributed over different classes
- 0-1 loss function can’t be used because it ignores the asymmetry of the cost between minority and majority class

Possible Solutions

- One way to handle cost function problem is to assign 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).

Continued...

- Another approach to deal with imbalance streaming data is to directly optimize target measures in an online fashion, 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).

Online Multiple Cost-Sensitive Learning

$X_t \in R^d$ denotes a feature vector received after t iterations

$Y_t \in \{1,-1\}$ denotes its true label

$f_t(x) : R^d \rightarrow R$ is a prediction function at the t-th iteration

$\mathbb{I}(b)$ indicator function

$\mathbb{I}(b) = 1$ if b is true else 0

$l(z) = \max(0, 1 - z)$ hinge loss function

Cost Function

$C(f(x), y) = c_{+}\mathbb{I}(y = 1)(f(x)) + c_{-}\mathbb{I}(y = -1)(-f(x))$

Where $C = (c_{+}, c_{-})$ is the cost vector that controls the balance between the two loss terms.

How to decide the value of C_+ and C_-

- Some approach use ad-hoc However, there is no guarantee that these ad-hoc approaches use appropriate values for c_+ and c_-.
- If the c_+ and c_- are changing during training itself, then it is difficult to analyse the performance of the classifier.
- Another approach is to use cross-validation that tunes the value of c_+ and c_- based on the offline separate validation set. However this approach is not feasible in online learning.

Online learning framework of multiple cost-sensitive learning.

The motivation is that if multiple classifiers with a number of c are learned simultaneously, there must exist one setting that is most appropriate to the data. Without loss of generality,

$c_+ + c_- = 1$ as a result only one parameter to be set c_+

continued...

To construct the pool of multiple values of c_+, we discretize (0, 1) into K evenly distributed values $\theta_1,...,\theta_K$, i.e., $\theta_j = j/(K + 1)$. With the value of $c+ = 1-\theta j/2$, the corresponding cost sensitive loss is denoted by

$$l_c^j(f(x), y) = (1 - \theta_j/2)l(y = 1)(f(x)) + (\theta_j/2)l(y = -1)(-f(x))$$

Sequences of classifiers $f_1^t(\cdot)$, $f_2^t(\cdot)$,...,$f_K^t(\cdot)$ simultaneously in online learning, with each sequence of $f_j^t(\cdot)$, t = 1,...,T to minimize the associated regret

$$R_T^j = \Sigma_{t=1}^{T} l_c^j(f_t^j(x_t), y_t) - \min_f \Sigma_{t=1}^{T} l_c^j(f(x_t), y_t)$$

How to choose a classifier from K classifiers

- 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.
- Proposed theoretically sound randomized method that selects a classifier for prediction according to a distribution $P_t = (p_t^1, p_t^2 \ldots\ldots, p_t^K)^T$ such that $\Sigma_{j=1} p^j = 1$

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

Continued...

where $\gamma > 0$ is a learning rate hyper-parameter, and M_t^j is some favorite performance measure (the higher the better) on historical examples (x_τ , y_τ), $\tau = 1,...,t-1$ using the predictions $f^j_1 ,...,f^j_{t-1}$ of the j-th sequence of classifiers.

Using the above equation, we can say that classifier with higher performance will have a higher probability of getting selected for making the prediction. This approach reduces to greedy approach when $\gamma \rightarrow \infty$.

OMCSL for F-measure

Given a sequence of labels y1,...,yt and a sequence of predictions f1,...,ft, 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 $\hat{y}_t = \mathbb{I}(f_t > 0)$

OMCSL for F-measure(conti.)

$$a_t = \sum_{\tau=1}^{t} \bar{y}_\tau \hat{y}_\tau$$

$$c_t = \sum_{\tau=1}^{t} \bar{y}_\tau + \sum_{\tau=1}^{t} \hat{y}_\tau$$

$$F_{t+1} = \frac{2a_t}{c_t}$$

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

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

OMCSL for AUPRC and AUROC

Let's define L_t^+ and L_-^t be two hash tables with length 'm' that partitions (0,1) in m partitions (0,1/m),(1/m,2/m),...((m-1)/m,m). For i $\in$ {1, . . . , 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 $\sigma(f) \in [(i-1)/m, i/m)$ and $L_t^-[i]$ stores the number of negative examples before the tth iteration (including t-th iteration) whose predictions f are such that $\sigma(f) \in [(i-1)/m, i/m)$.

Continued..

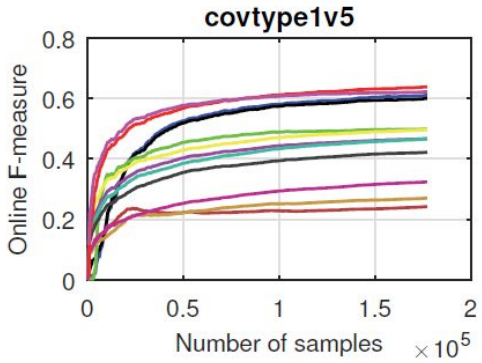
If $y_{t+1} = 1$, $\mathrm{AUROC}_{t+1} = \frac{N_t^+}{N_t^+ + 1}\mathrm{AUROC}_t + \frac{1}{(N_t^+ + 1)N_t^-}\left(\sum_{j=1}^{i} L_t^-[j] + L_t^-[i+1]/2\right)$, where i is the largest index such that $i/m \leq \sigma(f_{t+1})$.

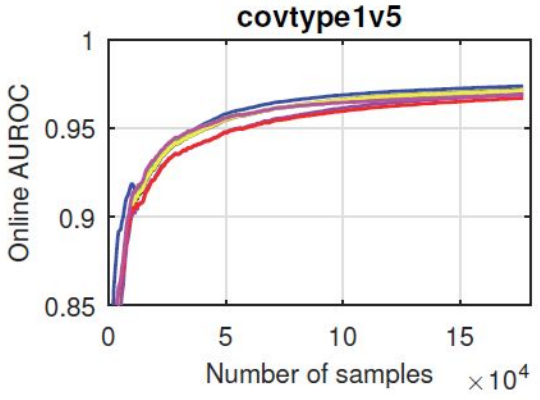
If $y_{t+1} = -1$, $\mathrm{AUROC}_{t+1} = \frac{N_t^-}{N_t^- + 1}\mathrm{AUROC}_t + \frac{1}{N_t^+(N_t^- + 1)}\left(\sum_{j=i+1}^{m-1} L_t^+[j] + L_t^+[i]/2\right)$, where i is the smallest index such that $i/m \geq \sigma(f_{t+1})$.

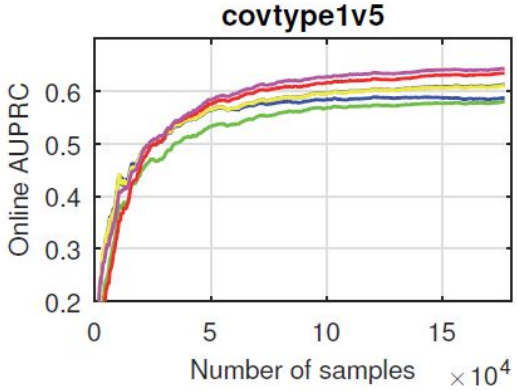
$\mathrm{AUPRC}_{t+1} = \frac{1}{2}\sum_{i=0}^{m-1}(\mathrm{R}(i) - \mathrm{R}(i+1))(\mathrm{P}(i) + \mathrm{P}(i+1))$, where

$$\mathbf{R}(i) = \frac{\sum_{j=i+1}^{m} L_t^+[j]}{N_t^+} \quad \text{and} \quad \mathrm{P}(i) = \frac{\sum_{j=i+1}^{m} L_t^+[j]}{\sum_{j=i+1}^{m} L_t^+[j] + \sum_{j=i+1}^{m} L_t^-[j]}$$

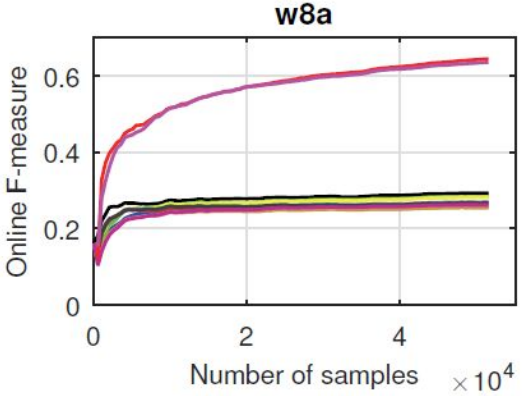
Result Comparison (covtype1v5 Dataset)

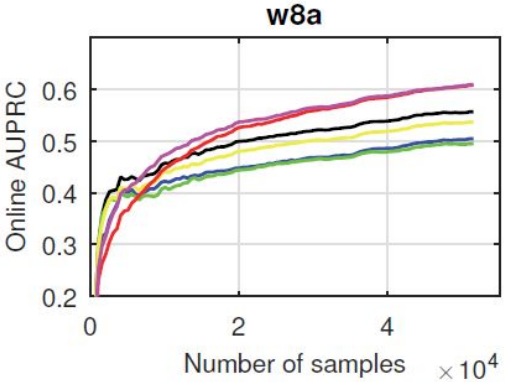


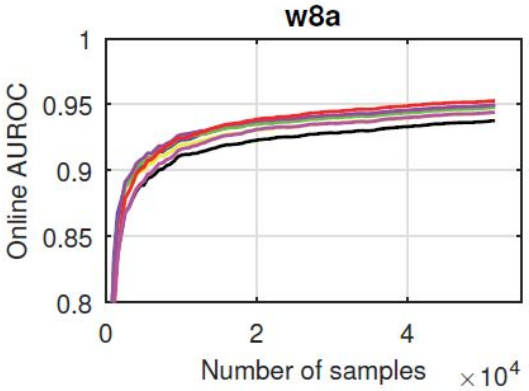




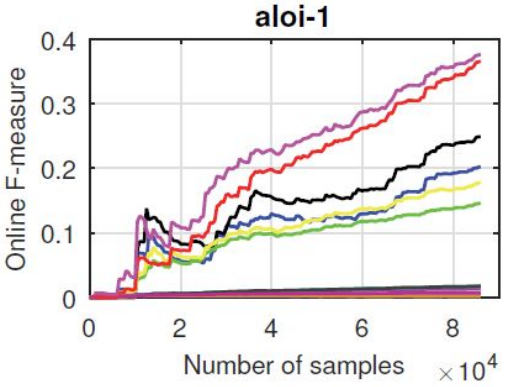
Result Comparison (w8a Dataset)

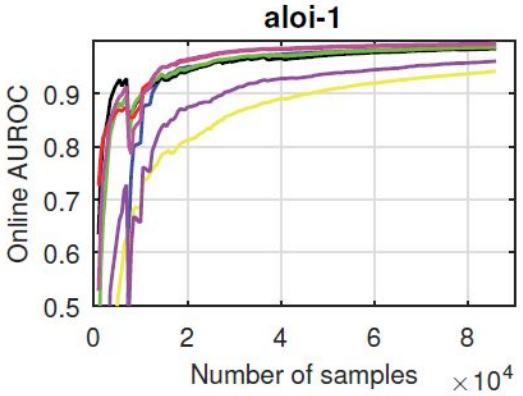


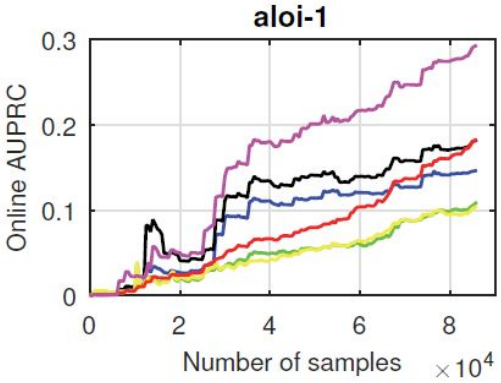




Result Comparison (aloi-1 Dataset)







Result Comparison

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**

Thank You