Online Learning and Active Learning

A comparative study with Support Vector Machine (SVM)

Ezukwoke K.I[1], Zareian S.J[2]

[1, 2] Department of Computer Science
Machine Learning and Data Mining
{ifeanyi.ezukwoke, samaneh.zareian.jahromi}@etu.univ-st-etienne.fr
University Jean Monnet, Saint-Etienne, France

Abstract

Passive aggressive online learning is an extension of Support Vector Machine (SVM) to the context of online learning for binary classification. In this paper we consider the application of the algorithm on IJCNN 2001 Neural Network Competition dataset from LibSVM dataset repository [1]. We also work on an improved version of the online learning algorithm called **Active learning** and we compare both algorithms to that of LibSVM. We then formalize and model the kernel versions of online and active learning algorithm wih experimental comparisons.

Keywords
Passive aggressive online learning, Active learning, Support Vector Machine (SVM).

1 INTRODUCTION

Online learning is a binary classification algorithm which is an extension of Support Vector Machine (SVM). Unlike SVM where we update the weights using batches of data samples or all data samples, here we only update the weights using one example at a time. At every iteration, we only update the weights associated with the data sample until we reach the last index.

[1]IJCNN 2001 NN Competition dataset LIBSVM

2 BINARY CLASSICATION

Binary classification is a classification problem involving only two defined categories. Given set of datasamples $\{x_i, y_i\}^N$, where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ where $\mathcal{Y} \in \{-1, +1\}$ for a binary classification problem.

2.1 Passive-Agrressive Online algorithm

Passive-Aggressive online algorithm is a family of first-order online learning algorithm. We present the optimization problem of the Passive Aggressive (PA) online algorithm from the original authors [1]. Online binary classification takes place in a sequence of rounds.

On each round the algorithm observes an instance and predicts its label to be either $+1$ or -1. After the prediction is made, the true label is revealed and the algorithm suffers an instantaneous loss which reflects the degree to which its prediction was wrong. At the end of each round, the algorithm uses the newly obtained instance-label pair to improve its prediction rule for the rounds to come [1].

We denote the symbols used to present the algorithm as follows. At round t t we select an example from an independent and

identical distributed random variable x_t with label $y_y \in \{-1, +1\}$. Base on the loss suffered we compute the update rule τ and update the weight w at every round unti we reach the final index. PA algorithms aggressively make an update whenever the loss is nonzero (even if the classification is correct) [2].

$$\begin{cases} \arg\min_{\mathbf{w}\in\mathbb{R}^N} \frac{1}{2}\|\mathbf{w} - \mathbf{w}_t\|^2 \\ s.t \quad y_t(\mathbf{w}\cdot\mathbf{x}_t) \geq 1 \end{cases} \tag{1}$$

where $\mathbf{w}$ is the weight update. By introducing the Lagrangian multipliers we have

$$\mathcal{L} = \frac{1}{2}\|\mathbf{w} - \mathbf{w}_t\|^2 + \tau(1 - y(\mathbf{w}\cdot\mathbf{x}_t)) \tag{2}$$

Observing the KKT optimality conditions we have that

$$\nabla\mathcal{L} = 0 \tag{3}$$

So that

$$\nabla_{\mathbf{w}}\mathcal{L} = \mathbf{w} - \mathbf{w}_t - \tau yx \tag{4}$$

$$\mathbf{w} = \mathbf{w}_t + \tau yx \tag{5}$$

Substituting equation (5) into (2) we have that

$$\mathcal{L}(\tau) = \frac{1}{2}\tau^2 y^2\|\mathbf{x}\|^2 + \tau(1 - y(\mathbf{w}\cdot\mathbf{x}_t)) \tag{6}$$

given that $y \in \{-1, 1\}$ we can rewrite the above as

$$\mathcal{L}(\tau) = \frac{1}{2}\tau^2\|\mathbf{x}\|^2 + \tau(1 - y(\mathbf{w}\cdot\mathbf{x}_t)) \tag{7}$$

$$\mathcal{L}(\tau) = -\frac{1}{2}\tau^2\|\mathbf{x}\|^2 + \tau(1 - y(\mathbf{w}\cdot\mathbf{x}_t)) \tag{8}$$

By differentiating with respect to τ we have that

$$\nabla_\tau\mathcal{L} = -\tau\|\mathbf{x}\|^2 + (1 - y(\mathbf{w}\cdot\mathbf{x}_t)) = 0 \tag{9}$$

where

$$\tau = \frac{1 - y(\mathbf{w}\cdot\mathbf{x}_t)}{\|\mathbf{x}\|^2} \tag{10}$$

The above equation is the hard margin formulation for passive aggressive algorithm.

By introducing an error term or slack variable, we can transform the hard margin formulation to a soft margin. We write the soft margin formulation of passive aggressive algorithm as

$$\begin{cases} \arg\min_{\mathbf{w}\in\mathbb{R}^N,\xi\geq 0} \frac{1}{2}\|\mathbf{w} - \mathbf{w}_t\|^2 + C\sum_{i=1}^N \xi \\ s.t \quad y_t(\mathbf{w}\cdot\mathbf{x}_t) \geq 1 + \sum_{i=1}^N \xi \\ \qquad \xi \geq 0 \end{cases} \tag{11}$$

Following the same procedure for deriving τ as in the case of the hard margin; we also have that,

$$\tau = min\left\{C, \frac{1 - y(\mathbf{w}\cdot\mathbf{x}_t)}{\|\mathbf{x}\|^2}\right\} \tag{12}$$

C is the positive parameter to balance the tradeoff between **passiveness** and **aggressiveness**.

Algorithm 1: Online Passive-Aggressive Algorithm

```
Input  : X, y, C ≥ 0
Output : w
begin
    w ← w^0;
    for t = 1, 2, ... N do
        receive instance: x ∈ R^n;
        predict ŷ = sign(w_t · x);
        correct label: y_t ∈ {−1, 1};
        loss
          l_t = max(0, 1 − y_t(w_t · x));
        compute τ_t;
        update: w_t ← w + τ y_t x_t
    end
end
```

where τ takes the different form

1. $\tau = \frac{l_t}{\|x_t\|^2}$
2. $\tau = min\left\{C, \frac{l_t}{\|x_t\|^2}\right\}$
3. $\tau = \frac{l_t}{\|x_t\|^2 + \frac{1}{2C}}$

2.2 Active Learning

The labeling of training examples is an expensive and time consuming task in a supervised classification setting. Active learning (AL) modelling is a semi-supervised strategy that helps to reduce labelling cost by selecting the most useful unlabelled examples to train a predictive model [3]. Reducing labelling cost helps to construct a high peforming classifier which keeps the amouunt of supervision to a minimum by selecting since it only selects the valuable training instances [4].

The promise of AL is that by iteratively increasing the size of our carefully selected labeled data, it is possible to achieve similar (or greater [5]) performance to using a fully supervised data-set with a fraction of the cost or time that it takes to label all the data. AL is considered to be a semi-supervised method, between unsupervised and fully supervised in terms of the amount of labeled data, i.e., for unsupervised data we use 0% labeled samples and for fully supervised we use 100% labeled samples. Therefore, the decision of how much data to use or alternatively how much performance is required from the model relies on a resource management decision, in other words it can be a business decision.

The active learning version of the algorithm 1's objective is to minimize the number of labels to query using a probabilistic criterion (**Bernoulli random distribution**). This way, the active learner aims to achieve high accuracy using as few labeled instances as possible, thereby minimizing the cost of obtaining labeled data.

Algorithm 2: Active Learning version of Algorithm 1

Input : $\mathbf{X}, y$
Output : $\mathbf{w}$
begin
  $\mathbf{w} \leftarrow \mathbf{w}^0$;
  for $t = 1, 2, \ldots, N$ **do**
    receive instance: $\mathbf{x}_t \in \mathbb{R}^n$;
    Let $\hat{p} = \mathbf{w}_t \cdot \mathbf{x}$;
    predict $\hat{y} = sign(\hat{p})$;
    Draw a Bernoulli random variable $Z_t \in \{0, 1\}$ of parameter $\delta/(\delta + |\hat{p}|)$;
    if $Z_t = 1$ **then**
      Get label $y_t \in \{-1, 1\}$;
      Loss $l_t = max(0, 1 - y_t(\mathbf{w}_t \cdot \mathbf{x}))$ compute τ_t;
      update: $\mathbf{w}_t \leftarrow \mathbf{w} + \tau y_t \mathbf{x}_t$
    else
      $\mathbf{w}_t \leftarrow \mathbf{w}$
    end
  end
end

3 KERNEL METHODS

3.0.1 Kernel Passive-Aggressive Online Learning Algorithm

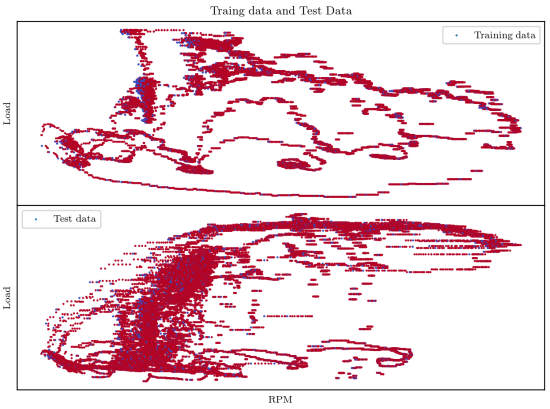
Kernel methods introduce non-linearity into our algorithm by projecting our data into a Hilbert space ($\mathcal{H}$). We implement the kernel passive-aggressive algorithm by considering that any classifier can be defined as a weighted sum of seen examples.

4 EXPERIMENT

4.1 Dataset

We used **ijcnn1** dataset from LIBSVM. These samples of data are produced by physical system (10-cylinder internal combustion engine). This time series dataset is labeled for sensors so that its normal firing is labeled by –1 and the misfire is labeled by +1, with

normals dominating. The models we introduce here try to do anomaly detection on this time series data.



The data is categorized into three different parts as training data, validation data and test data. The training data is a combination of training and validation datasets totalling 64980 datapoints. The training data is also composed of 58689 positive triggers (with labels -1) and 6291 anomaly triggers (with labels +1). The test data has a total 91701 data points with 82989 normal triggers and 8712 abnormal triggers. In general, each sample has 27 features however, we removed unneeded features and diminished it to 12 features.

4.2 Performance Aanlysis

In this experiment we compare the performances of passive aggressive algorithm and active passive aggressive aggressive algorithm with linear support vector machine from (**LibSVM**). We compare precisely their F1-score and ROC score using a well organized table as seen below.

Active Learning form of Online Learning algorithm is much more fast since It does not update on every iteration. It only updates the weights when a prediction is wrong. The time for Online Learning alorithm is 5.3 seconds, in contrast with Active Learning which is 0.83 seconds.

To compare Online Learning algorithm and its Active Learning version, in another point of view, the number of times Online Learning method made mistakes was 3190 in mode, with f1 relaxation, 3065 and with f2relaxation 3162. However the number of times Active Learning method made mistakes was 3147 in standard mode, with f1 relaxation, 3026 and with f2relaxation 3113.

We conclude then the number of mistakes made by Active Learning model is less than that of Online Learning model. Among the tree methods of weight updating, f1 relaxation performs better in both cases.

4.3 CONCLUSION

	AUCROC Score (%) | F1-score (%)							
	Classic		F1_relax, C = 0.1		F1_relax, C = 1		F1_relax, C = 10	
	AUC_ROC	F1-score	AUC_ROC	F1-score	AUC_ROC	F1-score	AUC_ROC	F1-score
PA	**83.85**	**67.64**	85.62	**68.38**	**86.22**	**69.56**	83.85	67.65
APA	71.72	47.50	**88.34**	65.33	79.54	59.05	**86.82**	**68.09**
SVM	-	-	61.97	36.73	63.19	39.42	63.32	39.70
	Classic		F2_relax, C = 0.1		F2_relax, C = 1		F2_relax, C = 10	
	AUC_ROC	F1-score	AUC_ROC	F1-score	AUC_ROC	F1-score	AUC_ROC	F1-score
PA			**85.15**	**69.05**	85.26	**69.19**	82.52	**66.39**
APA			84.25	63.47	**87.77**	68.54	**85.87**	65.07
SVM			61.97	36.74	63.19	39.42	63.32	39.70

Table 1: Comparing AUCROC and F-1 Scores for Passive-Aggressive (PA) algorithm, Active Passive-Aggressive (APA) algorithm and Support Vector Machine (SVM).

Classic Running Time (secs)		PA	APA	SVM	F1_Relax Running time (secs)		PA	APA	SVM	F2_Relax Running time (secs)		PA	APA	SVM
	C = 0.1	0.89	**0.52**	19.19		C = 0.1	1.01	**0.53**	19.19		C = 0.1	0.95	**0.53**	19.19
	C = 1	0.94	**0.53**	21.07		C = 1	0.95	**0.53**	21.07		C = 1	0.93	**0.53**	21.07
	C = 10	0.89	**0.54**	30.78		C = 10	0.94	**0.53**	30.78		C = 10	0.92	**0.53**	30.78

Table 2: Comparing Running time for Passive-Aggressive (PA) algorithm, Active Passive-Aggressive (APA) algorithm and Support Vector Machine (SVM).

References

[1] Crammer K., Dekel O., Keshet J., Shalev-Shwartz S., Singer Y. Online Passive-Aggressive Algorithms. *Journal of Machine Learning Research*, vol. 7, pp. 551–585, 2006.

[2] Hoi, S.C.H, Sahoo D., Lu, J., Zhao P. Online Learning: A Comprehensive Survey. *arXiv:1802.02871v2[cs.LG]*, pp.13-15, 2018.

[3] Djallel Bouneffouf, Romain Laroche, Tanguy Urvoy, Raphael Feraud, Robin Allesiardo. Contextual Bandit for Active Learning: Active Thompson Sampling. *Researchgate publication*, 2014.

[4] Liantao Wang, Xuelei Hu, Bo Yuan, Jianfeng Lu. Active Learning via Query Synthesis and Nearest Neighbour Search. *Neurocomputing*, 2014.

[5] Ilhan, Osman H., Amasyali M.F. Active Learning as a Way of Increasing Accuracy. *International Journal of Computer Theory and Engineering*. 6:(6), pp. 460, 2014.