Aggregating Algorithm

Andrew Robert Barraclough

Submitted for the Degree of Master of Science in

Machine Learning

Department of Computer Science
Royal Holloway University of London
Egham, Surrey TW20 0EX, UK

June 14, 2024

Declaration

This report has been prepared on the basis of my own work. Where other published and unpublished source materials have been used, these have been acknowledged.

Word Count:

Student Name: Andrew Robert Barraclough

Date of Submission: 29 August 2024

Signature:

Abstract

Your abstract goes here.

Contents

Acknowledgements

1 Introduction

The introduction should set the stage for the dissertation. It should provide background information on Online Prediction and the Aggregating Algorithm, highlihgt the importance of the study, and outline the research questions or hypotheses. Additionally, this section should briefly describe the structure of the dissertation.

2   On-line Machine Learning

2.1   Concept & Significance

Explain the fundamental concepts of on-line prediction and why it is significant in various fields such as finance, weather forecasting, and machine learning. Discuss its impact on real-time data analysis and decision-making.

The term "on-line machine learning" refers to a method of making real-time updates to models based on sequential data, which has significant impacts on fields such as finance, weather forecasting, and healthcare. In order to fully understand the concept, it is essential that we first have a good understanding of the fundamentals of Machine Learning and its applications in dynamic environments.

As Chair of Carnegie Mellon University's Machine Learning Department, Tom Mitchell, puts it, Machine Learning is a subset of artificial intelligence that "is a natural outgrowth of the intersection of Computer Science and Statistics" [5] that is concerned with developing algorithms and statistical models of the world that enable computers to perform tasks without the need for explicit commands, instead learning how to complete them purely from the data it is presented. Within this field, there exists many paradigms, each tailored to a specific problem type and dataset, including two that this report will focus on: "batch learning" and "on-line learning".

Unlike "batch learning", where the data that the algorithm is to learn from is readily available at the start of the training process, the order in which data is received by the model matters in "on-line learning" as it sequential. It is in this sequential nature of data that the significance of "on-line learning" lies; it allows for the continuous updating of models and predictions as new information arrives, without the need to completely retrain the model on the entire newly-updated training set.

The key principles and mechanisms of on-line prediction include its real-time nature, where data is processed as it arrives, allowing for immediate predictions and updates. This capability is crucial for applications that require instant feedback and decisions. On-line prediction involves learning on the fly, a process where models are updated incrementally with each new data point. This is in contrast to traditional batch learning, where models are trained on a fixed dataset all at once and used for predictions without further updates until retrained with new data. Common algorithms used in on-line prediction, such as stochastic gradient descent, online versions of

decision trees, and online support vector machines, are specifically designed to handle sequential data and update models incrementally. Evaluation of on-line prediction models typically employs metrics that can account for their adaptive nature, such as cumulative accuracy or regret.

When comparing on-line learning with batch learning, the differences are clear. Batch learning involves training models on a fixed set of data all at once, with predictions made using this static model until it is retrained with new data. On-line learning, however, continuously updates the model with each new data point, making it more adaptable and suitable for environments where data is constantly changing.

The advantages of on-line prediction are numerous. It offers adaptability, as models can quickly adjust to new information and changing patterns. It is efficient, eliminating the need for retraining models from scratch with new data. Moreover, it enables real-time decision-making, which is essential for applications requiring immediate responses, such as stock trading or fraud detection. However, on-line prediction also has its disadvantages. Implementing these systems can be technically challenging and resource-intensive, requiring significant computational power for continuous updating and processing. Additionally, without proper regularization, models may become overly sensitive to recent data points, leading to overfitting.

Understanding these fundamental concepts highlights the critical role of on-line prediction in various fields and its profound impact on real-time data analysis and decision-making.

- Define Batch Machine Learning

- Define On-line Machine Learning

- Highlight Differences between Batch & On-line

  "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." – *Mitchell, 1997*

2.2   Real-Time Decision-Making

Discuss how on-line prediction facilitates real-time decision-making processes. Provide examples of applications where immediate data processing and prediction are critical.

2.3 Challenges & Opportunities

Analyse the main challenges associated with on-line prediction, such as data volatility, computational limitations, and algorithmic efficiency. Highlight potential opportunities for advancements in the field.

2.4 Role in Prediction with Expert Advice

Describe how on-line prediction integrates with expert advice to enhance decision-making accuracy. Discuss the synergy between real-time data processing and expert algorithms.

An example of a reference:
[3], [4], [2], [6]

3   On-line Prediction

3.1   Preliminaries

3.2   Games

3.3   Prediction with Expert Advice

3.4   Specialist Experts

First introduced in the work of Freund et al. [1], 'specialist experts' can be thought of as an extension to the traditional on-line prediction framework that allows 'experts' to abstain from making predictions.

These experts are referred to as 'specialists' because they can be thought of as only making predictions "when the instance to be predicted falls within their area of expertise." In such cases where the expert is actively making a prediction, the expert is deemed to be 'awake', and is 'asleep' otherwise.

A prediction algorihtm may see that its internal confidence is low and decide to skip a turn in order to re-train. Alternatively, an algorithm may simply break down.

A natural idea for handling sleeping experts is to assume that a sleep expert "joins the crowd". Imagine that the sleeping expert sides with the learner and outputs the learner's prediction for that turn.

This modifies the Aggregating Algorithm, but that's to be discussed later!

In order to accommodate these 'specialist experts', we must modify the on-line prediction framework slightly. Similarly to the traditional framework, on-line learning with specialist experts can be thought of as a game that is played between a prediction algorithm, hereafter referred to as the 'learner', and an adversary, hereafter referred to as the 'nature'. The game is played in discrete iterations $t = 1, \ldots, T$, consisting of the same five steps:

1. The nature chooses a set of specialists that are 'awake' at iteration t, $E_t \subseteq \{1, \ldots, N\}$.

2. For each 'awake' specialist in the set chosen by the nature, $i \in E_t$, a prediction for that discrete iteration is output, $\hat{y}_i^t$.

3. The learner makes its own prediction based on the predictions of each awake specialist, $\hat{y}_t$.

4. The nature chooses an outcome y_t.

5. The learner suffers loss $\ell_L^t = L(\hat{y}_t, y_t)$.

6. The 'awake' specialists suffer loss $\ell_i^t = L(\hat{y}_t^i, y_t)$ while specialists that are asleep suffer no loss.

- We still assume that there are N 'experts', some of which may be 'specialist', indexed from $\{1, \ldots, N\}$.

- Predictions and Outcomes are real-valued numbers from a bounded range $[0, 1]$.

- Loss: $L : [0, 1] \times [0, 1] \to [0, \infty)$ associates a non-negative loss to each (prediction, outcome) pair.

To-Do

- How do we evaluate the performance of an algorithm?

  - In order to give the algortihm a meaningful bound, we compare the difference between the total loss of the algorithm and the total loss of the experts.
  - The total loss of the insomniac algorithm is compared against the loss of the best expert, but this doesn't make sense in this scenario because not all the experts are awake all the time, and may not make predictions.
  - Prove that the algorithm doesn't suffer large losses regardless of the adversary's strategy.

- How does this improve computational efficiency?

  - Discuss Problem Decomposition.
  - Naive algorithms make use of extremely large sets of experts which can make the calculation of predictions computationally expensive and infeasible.
  - By allowing for specialist experts, only a select handful of those experts make a prediction at a given time which can significantly reduce the computational load required.

- Discuss the applications of 'Specialist Experts'.

  - Markov Models
    * Talk about your theory and what your implemented code is about!
  - Switching Experts

3.4.1 Applications of Specialist Experts

4 Aggregating Algorithm

Describe the concept of the aggregating algorithm, its purpose, and how it synthesizes predictions from multiple experts to improve overall accuracy.

Algorithm 1 Aggregating Algorithm

1: initialise weights $w_0^i = q_i, i = 1, 2, \ldots, N$
2: FOR $t = 1, 2, \ldots$
3:      read experts' predictions $\gamma_t^i, i = 1, 2, \ldots, N$
4:      normalise the weights $p_{t-1}^i = w_{t-1}^i / \sum_{j=1}^N w_{t-1}^j$
5:      output $\gamma_t \in \Gamma$ satisfying for all $\omega \in \Omega$ the inequality
          $\lambda(\gamma_t, \omega) \leq -\frac{C}{\eta} \ln \sum_{i=1}^N p_{t-1}^i e^{-\eta \lambda(\gamma_t^i, \omega)}$
6:      observe the outcome ω_t
7:      update the experts' weights $w_t^i = w_{t-1}^i e^{-\eta \lambda(\gamma_t^i, \omega_t)}, i = 1, 2, \ldots, N$
8: END FOR

4.1 Weak Aggregating Algorithm

Explain the weak aggregating algorithm, its methodology, and its advantages. Discuss how it differs from stronger aggregating methods and its specific use cases.

4.2 Fixed Share Algorithm

Discuss the fixed share algorithm, its mechanics, and how it balances the use of different experts over time. Explain its relevance and application in dynamic environments.

4.3 Switching Experts

Analyze the strategy of switching between experts based on performance. Discuss the criteria for switching and its impact on prediction accuracy.

4.4 Specialist Experts & Sleeping Experts

Describe the role of specialist experts who focus on specific types of data or conditions. Discuss how their specialized knowledge enhances overall predictive performance.

Algorithm 2 Aggregating Algorithm for Specialist Experts (AASE)

1:   initialise weights $w_0^i = q_i, i = 1, 2, \ldots, N$

2:   FOR $t = 1, 2, \ldots$

3:       read the predictions, γ_t^n, of awake experts

4:       normalise the weights of awake experts
$$p_{t-1}^i = w_{t-1}^i / \sum_{i:E_i \text{ is awake}} w_{t-1}^i$$

5:       solve the system ($\omega \in \Omega$):
$$\lambda(\gamma, \omega) \leq -\frac{C}{\eta} \ln \sum_{n:E_n \text{ is awake}} p_t^n e^{-\eta \lambda(\gamma_t^n, \omega)}$$
        w.r.t. γ and output a solution γ_t

6:       observe the outcome ω_t

7:       update the awake experts' weights $w_t^n = w_{t-1}^n e^{-\eta \lambda(\gamma_t^n, \omega)}$

8:       update the sleeping experts' weights $w_t^n = w_{t-1}^n e^{-\eta \lambda(\gamma_t, \omega)/C(\eta)}$

9:   END FOR

4.5   Comparison with Model Selection

Compare the approach of prediction with expert advice to traditional model selection methods. Highlight the advantages and limitations of each approach.

References

[1] Yoav Freund, Robert Schapire, Yoram Singer, and Manfred Warmuth. Using and combining predictors that specialize. *Conference Proceedings of the Annual ACM Symposium on Theory of Computing*, 01 1997.

[2] Mark Herbster and Manfred K. Warmuth. Tracking the best expert. *Machine Learning*, 32:151–178, 1995.

[3] Yuri Kalnishkan. Prediction with expert advice for a finite number of experts: A practical introduction. *Pattern Recognition*, 126:108557, 2022.

[4] Yuri Kalnishkan and Michael V. Vyugin. The weak aggregating algorithm and weak mixability. *Journal of Computer and System Sciences*, 74(8):1228–1244, 2008. Learning Theory 2005.

[5] Tom M. Mitchell. The discipline of machine learning. 01 2006.

[6] Vladimir Vovk. Competitive on-line statistics. *International Statistical Review*, 69, 2001.