1 Introduction

A type of induction that combines training and testing of batch learning. This allows it to handle much more difficult inductive problems. However, it also makes a very strong assumption. After presenting the model, I will explain this assumption.

On-line learning is composed of a potentially infinite number of trials where each trial is broken down into three steps.

- Get the instance, $x_t \in X$.
- Predict the label of $x_t$. Call it $\hat{y} \in Y$.
- The environment returns the label, $y \in Y$.

The algorithm wants to minimize the number of mistakes, $\hat{y} \neq y$. (Other loss functions are possible.) Each trial the algorithm can use the information in $x$ and $y$ to refine its hypothesis. This model is very flexible and can allow things such as noise in the label, adversarial selection of instances, and shifting target functions. The strong assumption of on-line learning is that the algorithm always get label feedback at the end of a trial. This assumption is reasonable for problems where we predict the future.

2 Finite, No noise

- The inductive model with finite $H$ hypotheses.
- Mistake trees where adversary can pick instances.
- The halving algorithm makes at most $\lg H$ mistakes.
- The singleton set.
- The optimal algorithm.
- VC dimension bound.
• Threshold on $[0, 1]$ interval.
• Halving algorithm is somewhat intractable.

3 Countable, Noise

• WM bound and proof for finite $H$.
• Regret definition.
• WML tracking proof for finite $H$. Use minimum weight of $(\sum_{i=1}^{n} w_i)/(4n)$.
• WMI${}_2$ proof for countable $H$. Use starting weight of $1/t(t+1)$.
• Can randomized predictions (WMR) help?

4 Uncountable

• WMA with finite cover.
• WMA with many layers.
• Perceptron and Winnow are efficient and robust for learning half-spaces.

5 On-line to Batch

If the instances come from a fixed distribution and the on-line algorithm can make at most $M$ mistakes then with high probability, after $T$ trials, the average error rate of the hypotheses is close to $M/T$. If the average error rate were much higher then the algorithm would have a high chance of exceeding its mistake bound which is not possible. Formally, we can prove this average error rate depends on the observed number of mistakes. This is a data dependent bound.

Warning, just because an on-line algorithm has a good upper bound on mistakes does not guaranteed that the final hypothesis will have a low error rate. There are various techniques to convert the $T$ hypotheses generated during on-line learning into a good hypothesis for batch.

It is interesting to consider the martingale proof technique in the paper with what we already know about independent random variables.

• Independent random variables give exponential concentration.
• Correlation can break concentration.
• Not all correlation bad: Martingale difference property, $E(X_i|X_1, \ldots, X_{i-1}) = 0$.
• Some types of correlation can improve concentration.
  – Sampling without replacement.
  – Slot machine design.