Batch Performance for an Online Price

Koby Crammer, Mark Dredze, John Blitzer, Fernando Pereira
Computer and Information Science Department
University of Pennsylvania
Philadelphia, PA 19104
{crammer|mdredze|blitzer|pereira}@seas.upenn.edu

Batch learning techniques achieve good performance, but at the cost of many (sometimes even hundreds) of passes over the data. For many tasks, such as web-scale ranking of machine translation hypotheses, making many passes over the data is prohibitively expensive, even in parallel over thousands of machines [1]. Online algorithms, which treat data as a stream of examples, are conceptually appealing for these large scale problems. In practice, however, online algorithms tend to underperform batch methods, unless they are themselves run in multiple passes over the data [2].

In this work we explore a new type of online learning algorithm that incorporates a measure of confidence to the algorithm. The model maintains a confidence for each parameter, reflecting previously observed properties of the data. While this requires an additional parameter for each feature of the data, this is a minimal cost when compared to running the algorithm multiple times over the data. The resulting algorithm learns faster, requiring both fewer training instances and fewer passes over the training data, often approaching batch performance with only a single pass through the data.

Learning with Confidence

We begin with the mistake driven online learning framework for binary classification. On each round, the algorithm receives a new instance $x_t \in \mathcal{R}^n$, which is associated with an unknown label $y_t \in \{+1, -1\}$. The algorithm maintains a weight vector $w \in \mathcal{R}^n$ and makes predictions $\hat{y}$ as sign($w \cdot x$). The prediction $\hat{y}$ is returned and the correct label y_t is provided. If the prediction was incorrect, the algorithm updates w using x_t. A popular algorithm in this framework is MIRA [3], a margin infused passive-aggressive online algorithm similar to the Perceptron algorithm. MIRA updates in an aggressive manner, modifying w so that the prediction is correct with a margin.

To see intuitively how we could improve over MIRA, consider the text processing task of sentiment classification, where each instance corresponds to a product review and the algorithm must decide if the review is positive or negative. The problem is typically treated as a binary classification problem where reviews are represented by unigram and bigram features. Note that a non-zero feature value is an indication of a feature's appearance. Intuitively, the more a feature appears with a particular label, the more confident we can become in its weight.

MIRA does not take this into account. Imagine a positive book review containing the word "good" (the jth feature). MIRA correctly assigns some positive score to w_j. Suppose we see many examples containing the word "good", all of which are positive. Eventually, we will learn a weight for "good" that is highly positive, allowing us to correctly classify all of these instances with a margin. Now suppose that at some point in the future, we receive a negative example with the word "good" and a never before observed feature, "boring." If MIRA misclassifies this example, it will update w, decreasing the weight on "good" and "boring" equally. While the example is now correct, a better update would have been to only update the word "boring." We can see that it is this never before observed feature that lead to the mistake. A better update would favor changing the weight of this new feature, for which no evidence has been seen, over changing the feature "good" for which the algorithm has observed many instances.

To address this issue, we extend MIRA to include a confidence in learned weights that reflects previously observed instances. Each parameter of w has a corresponding confidence parameter, indicating the amount of knowledge or uncertainty the algorithm has in the current weight. Features that have been observed more often by the model will naturally have a higher confidence score. As the algorithm learns a new weight vector, it does so using the current confidence, updating the confidence accordingly.

Formally, the algorithm maintains an additional confidence vector $u \in \mathcal{R}^n$, where the value of u_i captures our confidence in the value of w_i. A high value of u_i reflects a high confidence in the value of w_i. Given a new labeled example (x_t, y_t) the algorithm makes a minimal modification to both w and u such that the prediction $\hat{y}$ will be correct even within the bounds of the uncertainty of the algorithm. Specifically, the output $\hat{y}$ should be correct even if the value of w_i will be modified in an amount proportional to u_i. The exact form of the update is omitted due to lack of space. We call this method **Variance** since the confidence vector u can be thought of as the uncertainty or variance of the associated weight.

Evaluation

We present some preliminary experiments to show the effectiveness of our algorithm. Both MIRA and the Variance method are evaluated on the task of sentiment classification, an example of a popular real world NLP task. We obtained data Amazon product reviews for seven domains and pre-processed the data as in [4]. A group of 2000 instances were randomly divided into 10 splits of 1600 train and 400 test instances. Each algorithm was trained on the training data for multiple iterations and evaluated on the test data after each iteration over the training data. The parameters w in both models were initialized to 0 and no regularization was used. Table shows the results for each method and domain evaluated on the test data after the first and fifth iteration. While MIRA improves significantly between the first and fifth iteration, Variance shows a smaller improvement, indicating that most learning was achieved in the first iteration. Additionally, Variance after a single iteration – the pure online setting – outperforms MIRA after 5 iterations – the batch setting. Variance effectively learns with a single pass over the data. This setting is explored using a larger amount of training data in the books domain. The same experiment was repeated using 10,000 training instances from books and 1,000 test instances averaged over 3 randomized runs. The model was evaluated after each training iteration on the test data. Results (figure 1a) show that even on a large amount of data, Variance attains near maximal performance after a single iteration while MIRA requires several passes over the data.

Next, we show that faster learning leads to improved online performance. The 10,000 unique book reviews were permuted into 10 orderings. Both MIRA and Variance were run in a single pass over all 10,000 instances. Error was measured as the average online error, ie. the cumulative mistakes divided by the total observed instances. This is also called the learning curve. The results in figure 1b, averaged over the 10 runs, show that Variance learns faster and produces an overall reduction in the number of online errors.

References

[1] Peng Xu. Personal communication, 2007.

[2] Vitor R. Carvalho and William W. Cohen. Single-pass online learning: Performance, voting schemes and online feature selection. In *KDD-2006*, 2006.

[3] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 2006.

[4] John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Association of Computational Linguistics (ACL)*, 2007.

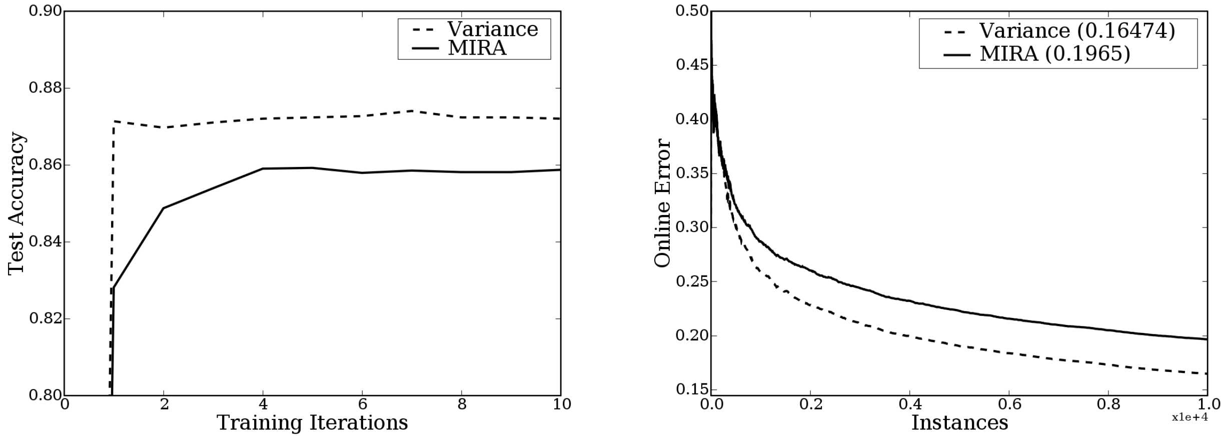

Figure 1: Left: The test accuracy of a classifier trained on 10,000 book product reviews after each iteration of training. Variance achieves high performance after a single pass while MIRA requires several iterations. Right: The online error of a single pass of both algorithms over 10,000 book product reviews with final error results in the legend. Variance learning learns faster on a single pass, leading to a reduced overall error.

Domain	*Algorithm*	*1 iteration*	*5 iterations*
apparel	MIRA	0.8479	0.8719
	Variance	0.8763	0.8820
books	MIRA	0.7813	0.8082
	Variance	0.8147	0.8192
dvd	MIRA	0.7728	0.7955
	Variance	0.8073	0.8150
electronics	MIRA	0.8065	0.8322
	Variance	0.8418	0.8458
kitchen	MIRA	0.834	0.8462
	Variance	0.8557	0.8570
music	MIRA	0.77	0.8060
	Variance	0.8158	0.8232
video	MIRA	0.7802	0.8095
	Variance	0.8192	0.8180

Table 1: Performance on 400 test instances of a classifier trained on 1600 instances from various domains for 1 and 5 iterations averaged over 10 trials. In each case, Variance performance after a single iteration improves on MIRA after 5 iterations.