Empirical Analysis of Supervised Learning Algorithms

Joshua Duquette * Justin Yang †

Abstract

We present an empirical observation of 5 different supervised classification algorithms, tested against four widely accepted binary classification problems taken from the UCI Machine Learning Repository. The model selection included SVM, Logistic Regression, Passive Aggressive Classifier, Random Forest, and K-Nearest Neighbors tuned against a pre-selected hyper-parameter grid search to optimize performance. The algorithms were evaluated on three metrics: f1-score, Matthew's correlation coefficient, and accuracy. We also gathered learning rate data to cross-compare efficiency metrics between algorithms as a secondary point of analysis.

Keywords: ICML, supervised-learning, empirical analysis

1 Introduction

While there are empirical studies comparing various supervised learning algorithms the primary goal of our research is to validate some of the results of the Caruana ICML as well as to expand it to include an additional algorithm, the passive-aggressive classifier. As the paper notes, different metrics give different insights into the performance of an algorithm on a given dataset so we will be looking at three different metrics, accuracy, f1-score, and Matthew's correlation coefficient as well as discussing which may be more representative given the algorithm and the data provided. Unlike the original research, however, we will not be considering the calibrating on the model's prediction using Platt Scaling nor Isotonic Regression. As a preview, our results seem to correlate very closely with the ICML document which is unsurprising as used the same procedures unless explicitly mentioned. While our research lacks some of the algorithmic and metric breadth, we feel that the larger number of iterations and retraining does add some weight to the analysis of our included algorithms. In addition, we feel that the inclusion of Matthew's correlation coefficient, especially in light of its robustness to unbalanced data, aids in providing a convincing and informative result.

2 Methods

2.1 Learning Algorithms

Our selection of algorithms was largely influenced by the original work of the Caruana ICML however we elected to add the passive-aggressive classifier (PAC) because of its similarity in construction to the SVM model: "Indeed, the core of our construction can be viewed as finding a support vector machine based on a single example while replacing the norm constraint of SVM with a proximity constraint to the current classifier." (Online Passive-Aggressive Algorithms, Crammer et al, 2006). As such we were interested in its performance against the SVM as the paper demonstrates that the SVM performs quite well for many datasets. Below we will be reviewing the parameter search space and any other algorithm-relevant information.

PROBLEM	#ATTR	TRAIN SIZE	TEST SIZE	%POZ
ADULT	14/105	5000	27561	24
COV_TYPE	54	5000	576012	49
LETTER.p1	15	5000	15000	4
LETTER.p2	15	5000	15000	50

TABLE 1: Description of problems

Support Vector Machine: Our search space included the linear, radial-basis function and polynomial kernel degree 2 and 3 for our optimization. In addition we also provided a C value of 1.e-07, 1.e-06, 1.e-05, 1.e-04, 1.e-03, 1.e-02, 1.e-01, 1.e+00, 1.e+01, 1.e+02, 1.e+03 to all kernels and a gamma of 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 2 for the radial basis function.

Logistic Regression: Our search space included three regularization functions (or lack-thereof) including: l1, l2 and no regularization. In addition we provided a C value of 1.e-08, 1.e-07, 1.e-06, 1.e-05, 1.e-04, 1.e-03, 1.e-02, 1.e-01, 1.e+00, 1.e+01, 1.e+02, 1.e+03, 1.e+04.

*Department of Computer Science, University of California, San Diego
†Department of Cognitive Science, University of California, San Diego

TABLE 2: Metric similarity analysis by classifier

TABLE 3: p-values

	KNN	Logistic	PAC	RF	SVM
KNN	0	0.000186	0.000081	0.261318	0.017856
Logistic		0	0.005986	0.000024	0.000089
PAC			0	0.000009	0.000039
RF				0	0.817689
SVM					0

TABLE 4: t-statistics

	KNN	Logistic	PAC	RF	SVM
KNN	0	4.178346	4.458043	-1.141729	-2.485636
Logistic		0	2.926512	-4.862667	-4.428268
PAC			0	-5.183160	-4.708541
RF				0	-0.232260
SVM					0

K-Nearest Neighbors: Our search space included included a geometric sequence for the number of neighbors ranging between 1 and 500 using 25 total subdivisions - notably this yielded a number of single neighbor searches. In addition we also provided the classifier weights uniform and distance. The algorithm that the KNN employed in this particular instance was 'ball tree' and featured a leaf size of 50 as per recommendation.

Random Forest: Our search space included a single selection for the number of nodes which we took from the Caruana ICML that suggested 1024. In addition we provided a max number of features as 1, 2, 4, 6, 8, 12, 16, 20.

Passive Aggressive Classifier: Our search space included a C value of 1.e-08, 1.e-07, 1.e-06, 1.e-05, 1.e-04, 1.e-03, 1.e-02, 1.e-01, 1.e+00, 1.e+01, 1.e+02, 1.e+03, 1.e+04. In addition we also provided to selections for loss function, namely hinge and square hinge loss.

All data was scaled using a min-max scalar between 0-1 as we make no assumptions about the distribution from which the data was selected. This is in contrast to the original Caruana ICML which apply a standard scalar for the logistic regression, KNN and SVM algorithms and do not apply a scalar for the remaining algorithms.

dataset	ACC	F1	MCC
letter_p1	0.962350	0.072568	0.0
letter_p2	0.503000	0.663995	0.0
adult	0.759190	0.388149	0.0
covtype	0.512401	0.655552	0.0

TABLE 5: Baseline Predicitons by Dataset

2.2 Performance Metrics

Our analysis only provides threshold metrics of which we selected three: accuracy, f1-score, and Matthew's correlation coefficient. The motivation for this selection was based on a novel analysis by BMC Genomics of these three metrics. The paper demonstrates that not all three metrics perform well when presented with datasets that include predominately balanced classes generating, as the paper says, "concordant scores", for data with many true-positive and true-negative predictions (therefore having fairly balanced classes for binary prediction). However, when presented with data that has many true-negatives but few true-positives of vice versa it demonstrates that "F1 and accuracy can provide misleading information, while MCC always generates results that reflect the overall prediction issues." (BMC Genomics, 2020). In addition, because MCC and f1-score as strongly related to their respective confusion matrix for the classifier, we felt that it was unnecessary to include given the used dataset.

2.3 Comparing Across Performance Metrics

All of our metrics follow the same rule that larger values indicate better performance, that being said their scalings nor their baselines are the same. The accuracy metric has a range of [0,1] while the f1-score and MCC (Matthew's correlation coefficient) have a range of [-1,1]. The accuracy and f1-score baseline is dependent on the data while the MCC is not. We have elected to include the baseline values for each metric relative to the dataset below for the purpose of comparing metrics across datasets.

Accuracy baselines were simply calculated by measuring the largest class against the size of the dataset and as MCC is data-independent its mean average value will always be 0.0. The f1-score proved to be somewhat more challenging as it isn't immediately evident how to empirically calculate a baseline. The solution we employed was to use a dummy predictor that always selected the positive class

		raw			normalized		
dataset	algorithm	ACC	F1	MCC	ACC	F1	MCC
adult	KNN	0.828598	0.507330	0.612841	0.705937	0.292717	0.409623
	Logistic	0.848397	0.562177	0.652493	0.722804	0.324362	0.436126
	PAC	0.839665	0.527269	0.611723	0.715365	0.304221	0.408876
	RF	0.850707	0.573323	0.665464	0.724772	0.330793	0.444796
	SVM	0.845881	0.554756	0.646761	0.720661	0.320080	0.432295
covtype	KNN	0.792618	0.585539	0.790082	0.652261	0.378500	0.649319
	Logistic	0.752906	0.506414	0.751371	0.619582	0.327352	0.617504
	PAC	0.735125	0.486404	0.754210	0.604950	0.314417	0.619838
	RF	0.821872	0.644257	0.820572	0.676335	0.416455	0.674376
	SVM	0.801670	0.604126	0.800954	0.659711	0.390514	0.658253
letter_p1	KNN	0.991822	0.884862	0.888423	0.985937	0.812848	0.818759
	Logistic	0.962222	-0.000769	0.000000	0.956513	-0.000707	0.000000
	PAC	0.962267	0.000000	0.000000	0.956557	0.000000	0.000000
	RF	0.988400	0.827323	0.823858	0.982535	0.759991	0.759256
	SVM	0.992511	0.895757	0.899585	0.986622	0.822855	0.829045
letter_p2	KNN	0.955622	0.911284	0.955425	0.915104	0.834110	0.915092
	Logistic	0.728533	0.457316	0.730549	0.697644	0.418587	0.699709
	PAC	0.730400	0.462582	0.724520	0.699431	0.423408	0.693935
	RF	0.946444	0.892906	0.946194	0.906315	0.817288	0.906251
	SVM	0.943867	0.887882	0.943967	0.903847	0.812690	0.904118

TABLE 6: Raw and Normed metric values presented for each dataset

and to measure it against the dataset. This provided mostly congruent results with the ICML with a notable departure when compared to the provided baseline for COV_TYPE. We suspect this is a result of a different dummy predictor being used although the paper doesn't elaborate on which so we elected to continue our approach for the sake of congruity.

To make the metrics cross-comparable we used the same schema as the original ICML by scaling the metrics between the baseline (thus allowing for negative performance if a model does worse than the baseline by chance) and the highest performing model score for a given metric.

2.4 Data Sets

We compare our algorithms across 4 binary classification problems taken from the UCI Machine Learning Repository: ADULT, COV_TYPE, and LETTER. Just as in the Caruana ICML we have converted the COV_TYPE by treating the largest class as the positive class and the rest as the negative class. In addition, we have split the LETTER dataset into LETTER.p1 and LETTER.p2 to produce a balanced and unbalanced dataset. The unbalanced dataset was achieved with a one-versus-all schema that set the letter 'O' as the positive class and the rest of the letters as the negative class. The balanced dataset was achieved by setting the first half of the letters, A-M, to the positive class and the remaining, N-Z, to the negative class.

3 Performances by Metric

For each dataset/algorithm combination, we randomly selected 5000 datapoints for training and reserved the rest as the test set for three iterations. We also one-hot encoded categorical columns to extract meaningful information without losing features (the original number of features along with the number of 'extracted' features is recorded in Table 1). This description is recorded below for ease of access.

Then, within each trial, we employed 5-fold stratified cross-validation on the 5000 randomly selected samples to obtain 5 inner trials, yielding a total of 15 results for each dataset/algorithm combo. These inner trials employed an 80-20 split for training and calibration of hyper-parameters After selecting the optimal parameters, we retrained the

model on the full training set of 5000 samples and evaluated the model on the test set. This was repeated thrice for each dataset/algorithm combo, yielding a total of 60 fully trained models.

Table 6 shows the mean test set scores for each dataset/algorithm combo (average of three repetitions). We evaluated the model's accuracy, f1 score, and Matthew's correlation coefficient to get a better idea of the model's true performance, and averaged the three metrics across iterations to produce the 'mean' values stored in said columns. In the table, we note that higher scores always indicate better performance, with optimal performance being a score of 1 for all metrics.

Across the three metrics, we see that KNN, SVM, and RF all performed similarly well but logistic regression and PAC struggled especially on both LETTER classification tasks. It does not appear to be the case, however, that these two algorithms performed poorly due to the unbalanced nature of these datasets: the %POZ rate of the ADULT task was 24%, but the two algorithms were comparable to the other 3 algorithms. Perhaps it has to do with the fact that both classifiers are linear models, whereas the other three can fit more complicated decision boundaries without over-fitting.

4 Performances by Problem

Table 6 shows the RAW and normalized scores for each of the 4 datasets broken down by metric. The entries are the mean validation score across the 3 iterations after training on the 5000 provided samples (this is done upon hyperparameter selection with 5-fold stratified kfold). The data suggests that the Random Forest and Support Vector Machine perform statistically better on each dataset that than either of the linear classifiers (PAC and LOGREG) and in some instances performs better than the KNN. Notably, the KNN performed worse on the ADULT dataset and COV_TYPE dataset but as well if not better on the LETTER.p1 and LETTER.p2 datasets. Statistically, the KNN's results are not significant compared the RF but are worse statistically compared to the SVM which can be observed in Table 3 and Table 4. Our p-value cutoff was selected for p > 0.05 to match the analysis done by the referenced ICML although a p > 0.01 makes no notable difference. Initially, we believed that the KNN may have performed better on data with balanced classes but this is evidently not the case as can be seen in Table 6. While the KNN is the best performing algorithm on LETTER.p2 (featuring a %POZ of .497) it performs worse than both RF and the SVM on the COV_TYPE dataset (featuring a similar %POZ of .487599). Instead, it is possible that the feature space of the data had the largest impact as the KNN performed best or relatively low feature count datasets. This is supported by KNN's generally poor performance on high dimensionality data although in some instances its performance is still not statistically worse than the aforementioned best performers.

dataset algorithm	adult	covtype	letter_p1	letter_p2	mean
KNN	0.649	0.722	0.921	**0.940**	0.808
Logistic	0.687	0.670	0.320	0.638	0.579
PAC	0.659	0.658	0.320	0.639	0.569
RF	**0.696**	**0.762**	0.879	0.928	0.816
SVM	0.682	0.735	**0.929**	0.925	**0.818**

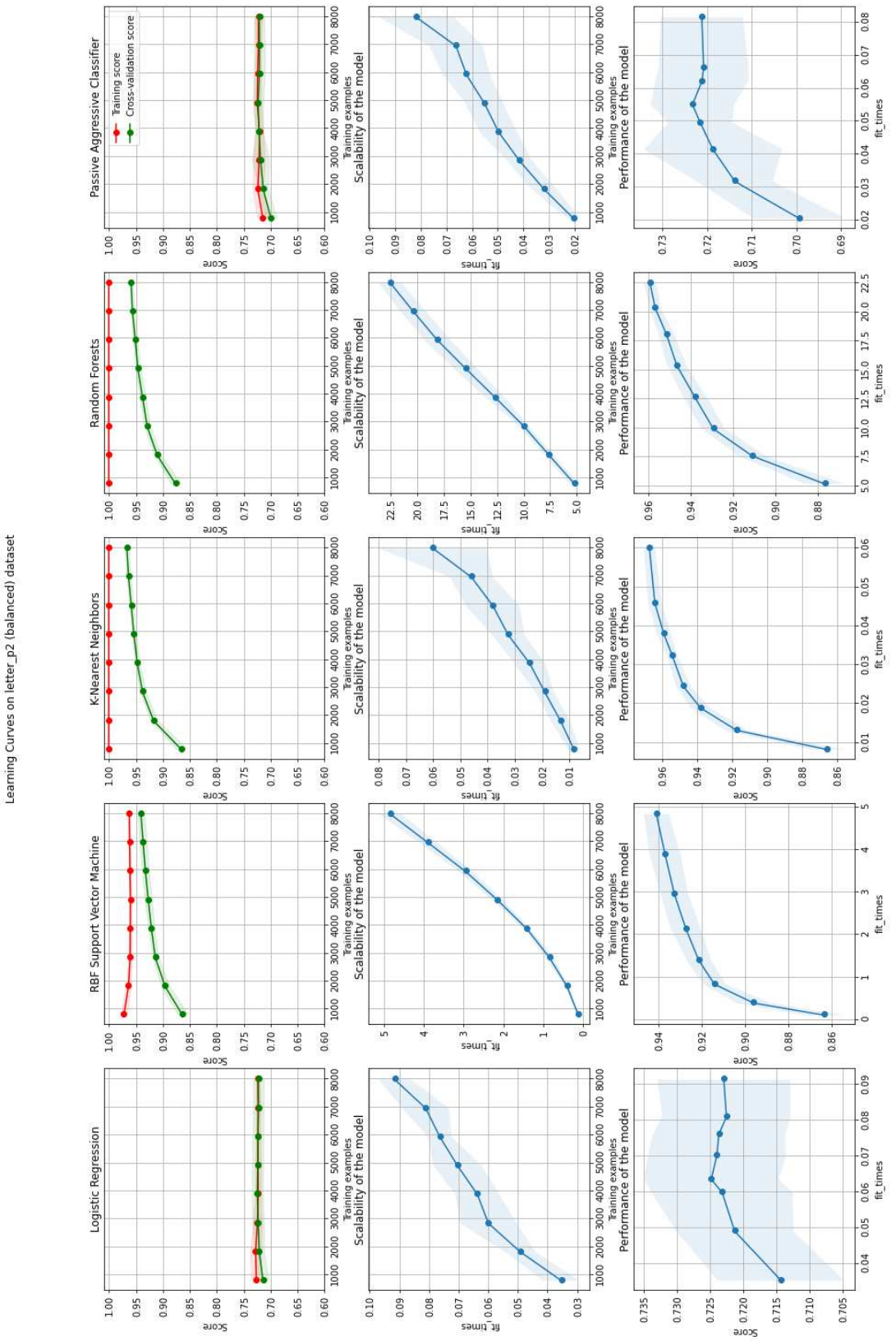
TABLE 7: Test set classifier performance for each dataset

algorithm	ACC	F1	MCC	mean
KNN	0.892165	0.722254	0.811693	0.808704
Logistic	0.823015	0.381284	0.533603	0.579301
PAC	0.816864	0.369064	0.522613	0.569514
RF	**0.901856**	0.734452	0.814022	0.816777
SVM	0.895982	**0.735630**	**0.822817**	**0.818143**

TABLE 8: Performance of classifier by metric

5 The Case for Bonus Points

While we did work in a group so it is expected that we would do more work we feel that amount of research, analysis and depth in this project does commensurate additional points. For instance, we provided 5 algorithms instead of the required 3. In addition, our selection of the Passive Aggressive Classifier was not an algorithm covered in class and required us to read its respective paper included in references to properly understand and analyze it. Further, we used 3 metrics instead of the required one and performed the necessary normalization and t-test analysis (all of the required code for this is provided in our support file). To supplement this we used a recent research paper that speaks directly to our metrics in our analysis which we also included in the references. In addition, we also provided supplementary analysis of these algorithms in the form of learning curve data as well as graphics for the validation by test for each algorithm. Next, we elected to use 4 datasets instead of the required 3 for our analysis by implementing the Caruana ICML's solution for acquiring unbalanced data as we were unable to locate


Learning Curves on letter_p2 (balanced) dataset
Logistic Regression
RBF Support Vector Machine
K-Nearest Neighbors
Random Forests
Passive Aggressive Classifier
Training score
Cross-validation score
Scalability of the model
Performance of the model
Training examples
Score
fit_times

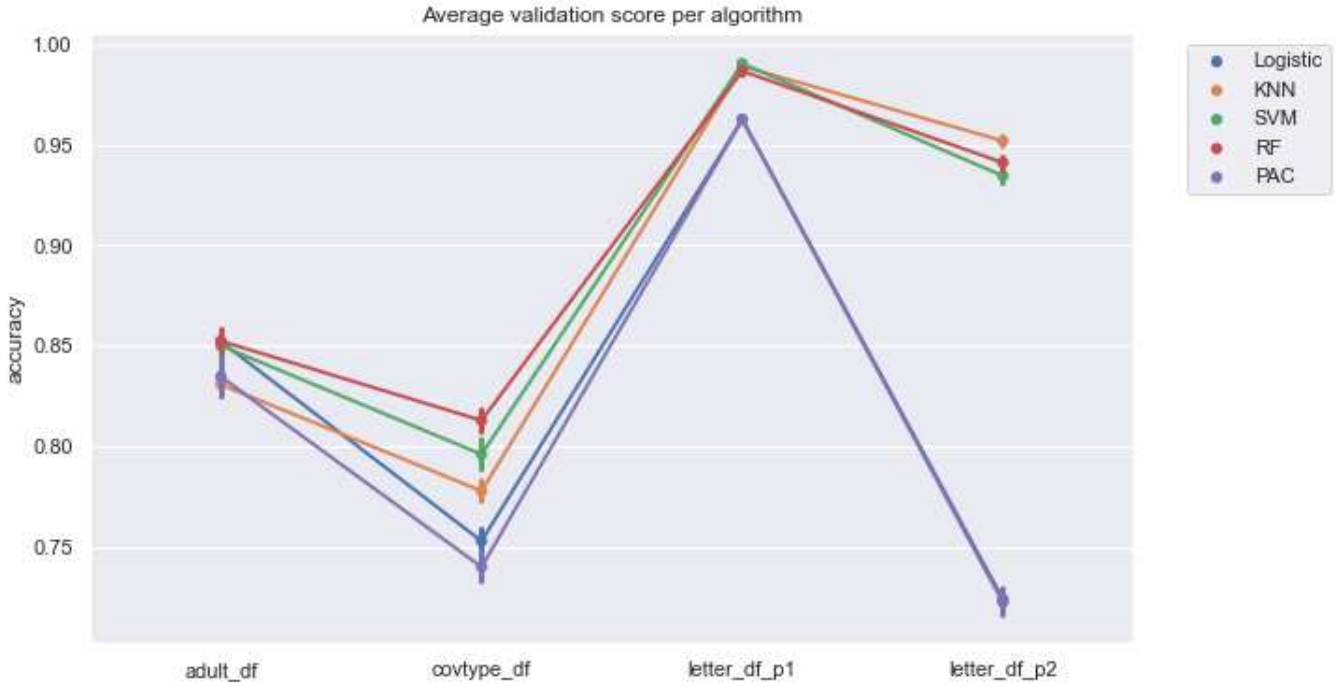


FIGURE 2: Validation scores for each dataset/algorithm combo, with error bars included.

data not on the Machine Learning Repository. Finally, our research is presented in the format of a formal research paper which required additional support code to render our tables and graphics in LaTeX as well as extensive LaTeX editing to emulate a formal research paper.

6 Conclusions

The Random Forest performed the best across all metrics for all datasets demonstrating that it is the best learning algorithm overall of those that we analyzed. The second best performer was the Support Vector Machine which performed statistically indistinguishable from the RF according to our paired-t-test analysis. To clarify, while our results numerically show that RF performed the best consistently, the difference between the SVM and RF is not statistically significant ($p \gg 0.05$) as can be seen in Table 3 and Table 4. The PAC performed the worst of all the algorithms and our t-test demonstrates that this was not due to chance alone ($p \ll 0.05$). Its redeeming quality was its speed, however, as it ran significantly faster than any other algorithm in our analysis. As noted in the Caruana ICML conclusion, many of these algorithms demonstrate significant variability, as can be seen in our validation accuracy graph, across datasets making it difficult to empirically classify one as better than the other. In addition, our learning rate analysis also demonstrates that metric performance isn't the only consideration to make in model selection especially with regard to speed and optimization iteration cutoffs. Perhaps another consideration to made is hyperparameter selection time as some algorithms have a much deeper pool of hyperparameters to optimize for.

References

Chicco, D., Jurman, G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics 21, 6 (2020).* https://doi.org/10.1186/s12864-019-6413-7

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, Yoram Singer Online Passive-Agressive Algorithms. *Journal of Machine Learning Research (3/2016).*

Rich Caruana, Alexandru Niculesc-Mizil, An Empirical Comparison of Supervised Learning Algorithms. *ICML '06: Proceedings of the 23rd international conference on Machine learning June 2006 Pages 161–168* https://doi.org/10.1145/1143844.1143865.