COMP 540: Statistical Machine Learning Final Project Proposal

Zekai (Jacob) Gao
jacobgao@rice.edu

1 Abstract

Traditionally, the runtime of training Support Vector Machines (SVMs) increases as the training set size increases. In this project, I will validate an opposite relevance. That is, the runtime of PEGASOS, a stochastic gradient descent optimizer for training linear SVMs, decreases as more samples are at hand.

2 Problem Statement

The goal of this project is to demonstrate inverse dependence between SVM optimization runtime and the training set size, as opposed to traditional runtime analysis of training SVMs. The inverse dependence here lies in a sub-gradient descent approach PEGASOS [5]. As a comparison, I will also run experiments on the same data using two traditional optimizers for training linear SVMs: the cutting planes method SVM-Perf [4] and the dual decomposition method SVM-Light [3].

The dataset I propose to use is the Reuters RCV1 collection [1], which contains 804,414 examples with 47,236 features. It is important that the dataset to be used is large enough so that the relevance could be accurately captured.

3 Motivation

The SVMs remain to be a popular research topic in recent years. Many approaches have been implemented and their time complexity studied. Most of the methods, such as SVM-Perf and dual decomposition, have their runtimes increasing as the training set size increases. In contrast, PEGASOS has no dependence on the training set size with its runtime $O(d/\lambda\epsilon)$, where d, λ and ϵ represent the feature dimensionality, regularization parameter and optimization accuracy respectively. Since the underlying target of SVM training is to find a classifier with low generalization error, we could achieve a fixed error rate with a certain number of examples. The excess examples could thus be used to optimize the predictor so that it will achieve the same error rate in less time.

Note that the three SVM methods mentioned above cannot be directly compared in runtime as they are dependent on different parameters. Nevertheless, I'm still interested in the unusual inverse dependence between runtime and training data size of linear SVMs using PEGASOS optimizer.

4 Hypothesis

As stated in [6], PEGASOS is the fastest published method for the RCV1 dataset. And it was even further optimized in efficiency by [8]. It could be shown that its runtime is not directly related with the training set size, making it scale better. Furthermore, for a stochastic gradient descent approach [5], as the sample size increases, the excess data could be used to decrease the runtime to get some desired optimization accuracy.

5 Method

For the empirical evaluations, I will fix the error rate, which may be the desired hinge loss or misclassification error, and then observe the runtimes of different SVM optimization methods under 4-6 training sets with increasing sizes, sampled independently from the RCV1 dataset.

6 Expected Results

As could be foreseen by now, the runtime of SVM training using PEGASOS will decrease as the training set size increases, while both SVM-Perf and SVM-Light will increase. Also I expect that PEGASOS is the fastest among the three.

7 Related Work

Many work has been done to improve the efficiency of SVM optimization and there exist many variants to obtain better runtimes with respect to certain parameters. For example, the dual-decomposition methods scale quadratically with training set size [2] while SVM-Perf scales linearly [7]. But SVM-Perf has much worse dependence on the optimization accuracy. PEGASOS is shown to handle this by scaling independently with training data size while still having the same dependence on the optimization accuracy with SVM-Perf [5]. In practice, both PEGASOS and SVM-Perf have good performance on large datasets with sparse linear kernels.

8 Timetable

Below is a timetable open to adjustment based on my progress:

- Feb 23: Project groups formed and project proposal finished.
- Mar 22: Finish running PEGASOS, SVM-Perf and SVM-Light on the RCV1 dataset. In the meanwhile, read papers on theoretical analysis on the runtimes of different SVM optimization methods.
- Apr 12: Prepare for final presentation and start writing final project report.
- Apr 25: Final project report finished.

References

[1] L. Bottou. Stochastic gradient descent examples. `http://leon.bottou.org/projects/sgd`.

[2] L. Bottou and C.J. Lin. Support vector machine solvers. *Large scale kernel machines*, pages 301–320, 2007.

[3] T. Joachims. *Making large-scale SVM learning practical*. MIT press, 1999.

[4] T. Joachims. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 217–226. ACM, 2006.

[5] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *Proceedings of the 24th international conference on Machine learning*, pages 807–814. ACM, 2007.

[6] S. Shalev-Shwartz and N. Srebro. Svm optimization: inverse dependence on training set size. In *Proceedings of the 25th international conference on Machine learning*, pages 928–935. ACM, 2008.

[7] A.J. Smola, SVN Vishwanathan, and Q. Le. Bundle methods for machine learning. *Advances in neural information processing systems*, 20:1377–1384, 2007.

[8] W. Xu. Towards optimal one pass large scale learning with averaged stochastic gradient descent. *Arxiv preprint arXiv:1107.2490*, 2011.