# Modeling SVM Parallelization

Igor Durdanovic, Eric Cosatto, Hans Peter Graf
NEC Laboratories, 4 Independence Way, Princeton, NJ 08540
igord@nec-labs.com; cosatto@nec-labs.com; hpg@nec-labs.com

We developed a general model for parallelizing Support Vector Machines that allows optimizing performance based on compute rate, memory size, communication bandwidth and communication latency. Experimentally this model was tested with up to 96 processors on a cluster of loosely coupled machines with problems of over 4 million training samples. Superlinear acceleration is obtained under most conditions, often by more than a factor of 10. Scaling to hundreds of machines is indicated by model simulations, way beyond what any other parallelization methods have achieved. Originally developed for handling the standard SVM optimization based on decomposition, the model has been extended to deal also with ridge-loss functions. This is particularly important for large problems because with a standard SVM algorithm the number of support vector grows rapidly with the number of training vectors. Using the ridge function, the number of support vectors can be reduced substantially in most problems.

Several parallelizations of the SVM algorithm have been proposed, and some have been demonstrated to handle problems of a few million training samples, such as the work described in [1],[2]. But the algorithms developed here scale to much larger numbers of processors and tolerate large communication latencies. This is significant because low-latency networks are very expensive, and low-cost clusters have inevitably a considerable latency in the communication. With this approach it becomes feasible to run problems with millions of samples on low-cost hardware without having to resort to approximations to the SVM algorithm or be restricted to certain types of kernels.

SVMs have always been hampered by a rapid increase of compute-time with the number of training vectors, limiting practical problem sizes to somewhere around 100,000 samples. As the results in table 1 demonstrate, problems with a few hundred thousand samples are solved in minutes and even 4 million samples take less than one day on a cluster of relatively outdated processors.

| Problem<br>Training size | Forest<br>522,765 | MNIST-E<br>222,411 | MNIST-E<br>508,368 | MNIST-E<br>1,016,736 | MNIST-E<br>2,033,472 | MNIST-E<br>4,066,944 |
|---|---|---|---|---|---|---|
| # SV | 50,375 | 25,106 | 41,875 | 63,074 | 97,455 | 196,977 |
| Time (sec.) | 765 | 288 | 780 | 6933 | 29021 | 64958 |
| Accuracy (%) | 98.34 | 98.58 | 98.85 | 99.12 | 99.27 | 99.13 |

Table 1: Summary of results obtained with a cluster of 48 dual Athlon (1.53GHz) machines. These are all two-class classification problems.

[1] Luca Zanni, Thomas Serafini, and Gaetano Zanghirati. Parallel software for training large scale support vector machines on multiprocessor systems. In Journal of Machine Learning Research, pages 1467–1492, 2006.

[2] Hans Peter Graf, Eric Cosatto, L´eon Bottou, Igor Durdanovic, and Vladimir Vapnik. Parallel support vector machines: The Cascade SVM. In Lawrence K.Saul, Yair Weiss, and L´eon Bottou, editors, Advances in Neural Information Processing Systems 17, pages 521–528. MIT Press, Cambridge, MA, 2005.