Parallel Implementations of Multiclass Support Vector Machines Optimized for Nonlinear Kernels

Matthew Barga
April 18 2012

Introduction

Advancements made in data storage technology within the last couple of decades have brought down costs and lead to an increase in large scale data retention. Many scientific and commercial organizations are maintaining large information databases specific to their needs. As a result of the continuing optimization of CPU architecture and random access memory technology, it has become feasible to implement computational numerical analysis on this archived data. The focus of the proposed research is the creation of efficient algorithms specifically targeted for large scale analysis through use of modern advances in computing architecture.

Classification algorithms constitute one family of numerical analysis method and are often used in pattern matching and machine learning applications. By properly training classification algorithms on data entities with a common set of features and a classification known a priori, classifiers are produced capable of accurately classifying new data entities with feature sets common to the training data, and for which a classification is desired. This kind of analysis makes it possible to easily extract information from sets of data based on complex relationships among data features that would otherwise be difficult to observe. One example of the application of these algorithms is in the prediction of medical risks and outcomes of a new patient. By training a classifier with a possibly extensive set of pre-specified medical attribute and case outcome data from records of previous patients, it can be used as an accurate predictor of new cases by inputting the values of the same set of medical attributes recorded for a new patient. This ability to classify based on a set of data features has proven useful in a wide range of applications including image and text processing (E. Osuna, E. OsunaE. Osuna1997b), drug discovery (Zernov, ZernovZernov2003), finance and e-commerce.

The Support Vector Machine, or SVM (Vapnik & Cortes, Vapnik & CortesVapnik & Cortes1995), is one type of classification algorithm that has enjoyed frequent use in recent years. The process of training SVMs still remains computationally and fiscally expensive as the processing speed and memory size needed to efficiently (in time) analyze large sets of data (i.e. $\gg$ 100,000 data entities) limits the amount of data that can practically be used for training (Burges, BurgesBurges1998). Training also involves a process of choosing different parameters, adjusting them and re-testing until satisfactory classification is achieved. These two factors limit the effectiveness of SVMs for use in real-time and industrial applications like finance, where it is necessary to process large databases of information and quick analysis is needed for decision making purposes.

Parallel processing is seen as a promising solution to this limitation. As many of the underlying computations needed for training can be done independent of each other, the work of training can be divided up among multiple processing units. As evidenced by Intel's abandonment of the next generation Pentium processor in favor of multi-core architecture in 2004, the future of computing lies in multi-core. In order to take full advantage of new parallel hardware coming into the market, new ways of implementing SVM algorithms need to be explored. Some previous methods that proved to be ineffective when executed on serial processors may translate very well to parallel processing platforms if they effectively divide up the data into independently computable portions. Finding and improving numerical methods for maximizing the accuracy of SVM algorithms is an area that is equally important to translating the algorithms to parallel architectures. It currently remains to be seen how to completely utilize nonlinear kernels efficiently on parallel systems.

Objectives

The core challenge of the proposed project is finding optimizations to improve efficiency of SVM training in large scale applications. Specifically, new methods for optimizing multiclass SVMs that utilize nonlinear kernels will be developed. There is an abundance of existing research that has focused on the creation of optimized training techniques for efficiently generating classifiers in a reasonable amount of time and with limited memory. The majority of these methods are implemented as sequential algorithms. When sequential algorithms are executed on parallel hardware without modification, there is no automatic performance gain. To take full advantage of multi-core hardware and other high performance computing resources available, restructured parallel versions of these algorithms are needed.

Runtime for multiclass SVM training scales even more poorly with input set size than for binary SVM training. Yet, there are many problems where classification into multiple categories is necessary. Most dominant multiclass classifiers work by decomposing a multiclass problem into multiple binary classification problems. Structures of this kind are well suited for both task and data level parallel optimizations. As multiclass techniques can benefit from multiple levels of parallel optimization and because their performance is particularly poor in large scale problems, they will be the focus of this project.

Existing training techniques will be selected based on potential for parallel optimization. The focus of interest is in techniques that will extend well to cases using nonlinear kernels in the SVM classifier. In current literature, methods that improved efficiency of training with linear kernels saw little or no improvement when used with nonlinear kernels. New numerical methods of training are neccessary to realize an improvement in the speed and accuracy of training with nonlinear kernels.

The goals of this project are concisely restated below:

- Identify training techniques well suited for use in multiclass classification with nonlinear kernel functions.
- Develop efficient numerical algorithms for classification with nonlinear kernel functions.
- Implement and optimize parallel algorithms implementing targeted optimization techniques.
- If techniques were applied to more than one multiclass SVM structure, compare performance among all structures that were optimized.
- Gather benchmarks and provide a comprehensive performance analysis.

Review of Background

The Support Vector Machine

The development and advancement of the SVM is often credited to Vladamir Vapnik who worked with related optimization theory in the 1980's and published a landmark paper outlining the first computational algorithm for an SVM in 1995 (Vapnik & Cortes, Vapnik & CortesVapnik & Cortes1995). Up to this point in time much attention has been paid to sequential optimization of SVM algorithms (Vapnik, VapnikVapnik1982; E. Osuna, E. OsunaE. Osuna1997a; Platt, PlattPlatt1999; Fan, FanFan2005). The current performance bottleneck lies in the computation needed to find a maximized classifying *margin*, a process which reduces to a quadratic programming optimization problem. The optimized methods cited above focus on reducing the size of the matrix in the quadratic programming problem and breaking up the problem into a number of smaller independent problems that have input matrices capable of fitting into local memory (Bishop, BishopBishop2007). Even using these optimizations, SVMs continue to suffer from performance issues stemming from memory size limitations and the data intensive computation involved, especially in large scale problems.

Although they were originally developed as binary classifiers, a number of SVM schemes capable of assigning an input to one of multiple classes have been developed (Vapnik, VapnikVapnik1998; Weston & Watkins, Weston & WatkinsWeston & Watkins1998). The most popular approach is to rely on the fundamental binary classifier as the underlying computational engine. In order to use binary classifiers, a multiclass problem is broken down into multiple binary classification problems. These methods of reduction to binary classification problems are typically of four main types, including one-vs-all, one-vs-one, error-correcting code and all-at-once classifiers (Abe, AbeAbe2005). Multiclass SVMs compound the performance issue as not only do potentially large data sets need to be analyzed, but multiple classifiers may need to be constructed for different class pairings and multiple comparisons may need to be performed to compare these generated classifiers against each other.

Optimizing for Parallel Architecture

Vapnik himself first proposed a technique called *chunking* (Vapnik, VapnikVapnik1982) which breaks up the full quadratic programming problem into a series of smaller problems. In 1996, *decomposition methods* (E. Osuna, E. OsunaE. Osuna1997a) further reduced the size of data sets necessary to consider for a series of smaller quadratic optimization problems to a fixed size that can fit in local memory for any arbitrarily large problem domain. *Sequential minimal optimization* (SMO) (Platt, PlattPlatt1999) remains one of the most common methods in practice today. SMO works by breaking up the quadratic programming problem into the smallest possible problem set sizes. Properties of these problems allow them to be solved analytically, eliminating the need for time-consuming numerical computation that relies on matrix multiplication (Platt, PlattPlatt1999).

Building off of these optimizations there are some natural steps that have been taken towards parallelization. In methods that break up the data into smaller sets, computations performed on each set can be done independently, and thus can be completed on separate computing resources. This has been explored for the SMO method (Cao, CaoCao2006). Work by Zanni et al. (Zanghirati ZanniZanghirati Zanni2003; ZanniZanni2006) also focused on creating a parallel version of SVM training based on a variant of the decomposition technique. In order to better understand the potential of parallel programming to optimize training techniques, new outlets for applying it need to be explored.

Although there have been efforts to create more efficient multiclass classifiers (Rifkin & Klautau, Rifkin & KlautauRifkin & Klautau2004; Crammer & Singer, Crammer & SingerCrammer & Singer2002; Bre-

densteiner & Bennett, Bredensteiner & BennettBredensteiner & Bennett1999; Hsu & Lin, Hsu & LinHsu & Lin2002), there has been relatively little implementation of parallel algorithms in optimization. Multiclass SVMs stand even more to gain from parallel optimization as parallelism can be exploited on multiple levels. Recent investigation into a parallel implementation of a one-vs-all scheme classifier produced an algorithm with computational complexity that was similar to the training of one of its underlying binary classifiers (Niu, NiuNiu2011). When the method in this project was extended to nonlinear kernels, the same performance gain was not observed. As nonlinear kernels can provide higher classification accuracy in a number of problems, research on the application of parallelism in creating efficient training techniques using nonlinear kernels is an important future topic.

Research Design and Methodology

Tasks & Experimentation

Expanding on currently available documented research, parallel implementations of efficient multiclass SVM classifiers will be developed for cases involving nonlinear kernels. The first step in this project will be deciding on suitable optimization techniques to be used for creation of the multiclass SVMs. Following a literature review, techniques will be chosen based on usefulness to multiclass classification and a stronger consideration will be given to promising methods that have not yet seen implementation and testing in current research. It will also be of interest to find a classifier that is efficient in the case of using nonlinear kernels. In order to achieve this, new numerical methods for efficiently training SVM classifiers on data will have to be developed. This alone will constitute much of the real work in the project.

A degree of freedom lies in choosing a platform to target for development of the parallel algorithms. There are multiple hardware platforms available that support parallel software. In modern high performance computing applications, cluster and supercomputing resources tend to be the most popular and widely available. The language independent *Message Passing Interface* (MPI) protocol is a commonly used tool to program parallel applications for distributed memory computing clusters. Due to its high portability, the large knowledge base available and its popular use in current SVM and other parallel literature, it will be chosen as the method of implementation for the work discussed. In order to run the algorithms once they are ready for execution and testing, access will be needed to high performance computing resources. A benefit of using MPI is that there is no specific requirement on the language used to write the algorithms and no specific computing resource necessary for implementation. Most large universities have access to some form of cluster computing or supercomputing resource and systems such as these would be well suited for deployment of the parallel algorithms to be developed.

Once suitable training techniques are chosen, development of the multiclass SVM algorithms will begin. The focus will be on finding structures in the chosen training methods that can be exploited for data and task level parallelism. The techniques developed may be applied to more than one multiclass SVM structure. The number of structures studied will depend on time constraints. An appropriate amount of time will also be allocated following the development of the algorithms for detailed testing and debugging of any code written. Upon satisfactory completion of the algorithms, final testing will be completed and benchmark specifications will be produced. A number of public databases for multiclass problems are available for benchmark testing purposes.

Qualifications

I first gained indirect exposure to classification algorithms in past research projects involving topics in mathematical modeling and machine vision. From 2008 to 2009, I participated in a research exchange program at Tohoku University with partial funding from a JASSO Student Exchange Support Scholarship. Under the supervision of Professor Ayumi Shinohara, I developed a basic face detection algorithm that combined several basic classification algorithms taking haar-like features (Viola & Jones, Viola & JonesViola & Jones2001) as input. I have also completed study as part of my undergraduate coursework in parallel architecture and in compiler design. Learning to recognize data dependence and coherence issues were important outcomes of these courses. These are key concepts that are crucial to successfully utilizing parallel programming. As the final project of the parallel architecture course, I successfully developed a simple shared memory dual-core processor on a field-programmable gate array (FPGA) for the MIPS instruction set architecture (ISA), and wrote multiple parallel algorithms for execution on the processor. It is my goal in this proposed project to combine my knowledge of parallel programming and machine learning constructs in order to create efficient SVM algorithms that will be of practical use in a number of large scale and data intensive problems in industry including image processing and real-time video applications.

Projected Timeline

Study	**Month-Year**	**Description of Work**
Phase One:	10-2012	Continue investigation of optimization techniques
Literature Review &	11-2012	
Study of MPI Protocol	12-2012	
	1-2013	
Phase Two:	2-2013	Begin investigation of nonlinear kernel techniques
Algorithm Development	3-2013	
	4-2013	
	5-2013	
	6-2013	
	7-2013	
	8-2013	Begin preliminary writing and testing of algorithms
	9-2013	
	10-2013	
	11-2013	
Phase Three:	12-2013	Begin detailed testing and debugging of algorithms
Testing and Debugging	1-2014	
	2-2014	
	3-2014	
	4-2014	
	5-2014	Algorithms complete
Phase Four:	6-2014	Further testing and performance analysis
Performance Analysis &	7-2014	
Completion of Thesis	8-2014	
	9-2014	

Deliverables

- Optimized parallel algorithms implementing multiclass SVMs *(May 2014)*
- Performance analysis comparing benchmark statistics among any SVMs developed in this work, and of SVMs in this work against sequential and parallel multiclass SVMs in field use *(July 2014)*
- Thesis report documenting design, implementation and testing of produced algorithms *(August 2014)*

References

Abe, S. (2005). *Support vector machines for pattern classification.* Springer.

Bishop, C. M. (2007). *Pattern recognition and machine learning (information science and statistics)* (1st ed. 2006. Corr. 2nd printing ed.). Springer.

Bredensteiner, E. J., & Bennett, K. P. (1999). Multicategory classification by support vector machines. *Computational Optimization and Applications*, *12*, 53-79. (10.1023/A:1008663629662)

Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, *2*, 121-167. (10.1023/A:1009715923555)

Cao, e. a. (2006). Parallel sequential minimal optimization for the training of support vector machines. *IEEE Transactions on Neural Networks*, *17*(4), 1039 - 1049.

Crammer, K., & Singer, Y. (2002). On the algorithmic implementation of multiclass kernel-based vector machines. *J. Mach. Learn. Res.*, *2*, 265–292.

E. Osuna, e. a. (1997a). *Support vector machines: Training and applications* (Tech. Rep.).

E. Osuna, e. a. (1997b). Training support vector machines: an application to face detection. In *Computer vision and pattern recognition* (pp. 130–136).

Fan, e. a. (2005). Working set selection using second order information for training svm. *Journal of Machine Learning Research*, *6*, 1889–1918.

Hsu, C.-W., & Lin, C.-J. (2002). *A comparison of methods for multi-class support vector machines.*

Niu, L. (2011). Parallel algorithm for training multiclass proximal support vector machines. *Applied Mathematics & Computation*, *217*(12), 5328 - 5337.

Platt, J. C. (1999). Fast training of support vector machines using sequential minimal optimization. In (pp. 185–208). Cambridge, MA, USA: MIT Press.

Rifkin, R., & Klautau, A. (2004). In defense of one-vs-all classification. *J. Mach. Learn. Res.*, *5*, 101–141.

Vapnik, V. (1982). Estimation of dependences based on empirical data.

Vapnik, V., & Cortes, C. (1995). Support-vector networks. *Machine Learning*, *20*, 273 - 297.

Vapnik, V. N. (1998). *Statistical Learning Theory.* Wiley-Interscience.

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In (pp. 511–518).

Weston, J., & Watkins, C. (1998). *Multi-class support vector machines.*

Zanghirati, G., & Zanni, L. (2003). A parallel solver for large quadratic programs in training support vector machines. *Parallel Comput.*, *29*, 535–551.

Zanni, e. a. (2006). Parallel software for training large scale support vector machines on multiprocessor systems. *Journal of Machine Learning Research*, *7*(7), 1467 - 1492.

Zernov, e. a., Vladimir V. (2003). Drug discovery using support vector machines. the case studies of drug-likeness, agrochemical-likeness, and enzyme inhibition predictions. *Journal of Chemical Information and Computer Sciences*, *43*(6), 2048-2056.