Scaling-up Quadratic Programming Feature Selection

Yamuna Prasad and **K. K. Biswas** and **Parag Singla**
Indian Institute of Technology Delhi
Hauz Khas, New Delhi INDIA 110016

Abstract

Domains such as vision, bioinformatics, web search and web rankings involve datasets where number of features is very large. Feature selection is commonly employed to deal with high dimensional data. Recently, Quadratic Programming Feature Selection (QPFS) has been shown to outperform many of the existing feature selection methods for a variety of datasets. In this paper, we propose a Sequential Minimal Optimization (SMO) based framework for QPFS. This helps in reducing the cubic computational time (in terms of data dimension) of the standard QPFS to quadratic time in practice. Further, our approach has significantly less memory requirement than QPFS. This memory saving can be critical for doing feature selection in high dimensions. The performance of our approach is demonstrated using three publicly available benchmark datasets from bioinformatics domain.

Introduction

Recently, a new filter based quadratic programming feature selection (QPFS) method has been proposed (Rodriguez-Lujan et al. 2010). Here, a similarity matrix which represents the redundancy among the features and a feature relevance vector are computed. These together are fed into a quadratic program to get a ranking on the features. To deal with the quadratic complexity, Rodriguez-Lujan et al. (2010) combine a Nyström sampling method, which reduces the space and time requirements at the cost of accuracy.

The complexity of learning Support Vector Machines (SVM) using quadratic program solver is cubic. Sequential minimal optimization (SMO) based decomposition significantly reduces the complexity of learning in SVMs (Guyon and Elisseeff 2003). In the SMO based approach, a working set of size two (i.e. two variables which most violate the optimality condition) is selected iteratively and the target function is optimized with respect to them.

In this paper, we propose an SMO type decomposition based on second order approximation for QPFS. We refer to our approach as QPFS-SMO, henceforth. We derive the conditions for selecting the working set for our formulation. Our proposed approach has computational time quadratic in the number of features in practice. This is in contrast to the cubic time complexity of QPFS. Our approach is also significantly more memory efficient. This time and memory saving can be critical for doing feature selection in high dimensional data where QPFS runs out of memory. Our experiments on three publicly available benchmark microarray datasets validate that QPFS-SMO is orders of magnitude faster, and significantly more memory efficient, than QPFS and QPFS with Nyström method, while retaining the same level of accuracy.

We first describe our SMO based formulation for QPFS. This is followed by experimental evaluation of the two approaches on the three datasets.

QPFS

Given a dataset with M features $(A_i, i = 1, ..., M)$, C class labels $(c_i, i = 1, ..., C)$ and l training instances $(x_i, i = 1, ..., l)$, the standard QPFS formulation (Rodriguez-Lujan et al. 2010) is:

$$f(\alpha) = \min_{\alpha} \frac{1}{2}(1-\theta)\alpha^T Q \alpha - \theta s^T \alpha$$
$$\text{Subject to} \qquad \alpha_i \geq 0, i = 1, ..., M; \qquad I^T \alpha = 1. \tag{1}$$

where, α is an M dimensional vector, I is the vector of all ones and Q is an $M \times M$ symmetric positive semi-definite matrix, which represents the redundancy among the features. s is an M dimensional vector of non-negative values, which represents relevance score of features with the class labels. In this formulation, quadratic term captures the dependence between each pair of features, and linear term captures the relevance between each of the features and the class labels. The scalar quantity $\theta \in [0, 1]$ represents the relative importance of non-redundancy amongst the features and their relevance. Rodriguez-Lujan et al. (2010) provide a detailed description of QPFS.

The Proposed QPFS-SMO Approach

It is easy to see that Equation (1) differs from the SVM formulation (Fan, Chen, and Lin 2005) only in the way constraints are expressed over α_i's. The primary difference lies in the constraint set and the feature relevance vector s. In SVMs, a vector of *ones* is used instead of the feature relevance vector s. The key component in the SMO type decomposition is to select a working set which maximally descends the objective value at each iteration. Following Fan, Chen, and Lin (2005)'s work for SVMs, we have developed

a second order approximation for working set (*two element*) selection for QPFS-SMO. After computing α vector, the features are ranked as done by Rodriguez-Lujan et al. (2010). Algorithm 1 summarizes our approach. Details are available in the supplementary material (Prasad, Biswas, and Singla 2013).

Algorithm 1: Proposed QPFS-SMO Approach

Input: Dataset, Value of θ parameter
Output: Solution vector α

1. Compute similarity matrix Q and relevance vector s. Scale Q and s by $(1-\theta)$ and θ, respectively.
2. Initialize α^1 to some feasible solution.
3. Set $k \leftarrow 1$.
4. Select working set $B = \{i, j\}$.
5. Set α^{k+1} to be the optimal solution.
6. Set $k \leftarrow k + 1$.
7. If α^k satisfies the stopping criteria, then exit. Otherwise, go to step 4.

Experiments

Datasets

We experiment with three publicly available benchmark microarray datasets, namely Colon, SRBCT and GCM (Rodriguez-Lujan et al. 2010; Ganesh Kumar et al. 2012). The number of features in these datasets are 2000, 2308 and 16063, respectively. The number of samples are 62, 63 and 190, respectively.

Methodology

We follow the methodology of Rodriguez-Lujan et al. (2010) for our experiments. Each dataset is divided into 90% and 10% sized splits for training and testing, respectively. The reported results are averaged over 100 random splits of the data. We use mutual information for redundancy and relevance measures. The data is discretized using three segments and one standard deviation for computing mutual information. The value of scale parameter (θ) is computed as described in Lujan et al. (2010). Linear SVM (L2-regularized L2-loss support vector classification in primal) (Fan et al. 2008) is used to train a classifier using the optimal set of features output by each algorithm. All the experiments were run on a machine with 16 processors (3.10 GHz) using 128 GB of RAM.

Results

Time and Memory Table 1 shows the time and memory requirements for feature selection done using QPFS and QPFS-SMO. On all the datasets, QPFS-SMO is faster than QPFS. On Colon and SRBCT, it is an order of magnitude faster. QPFS ran out of memory for the GCM dataset in contrast to QPFS-SMO which had no issue running on this dataset. On GCM, we also compared the performance of QPFS-SMO+Nyström with QPFS+Nyström at a sampling rate of $\rho = 0.03$. QPFS-SMO+Nyström (59.2 seconds) is about twice as fast as the QPFS+Nyström (97.5 seconds), while performing marginally better.

QPFS-SMO requires significantly less memory compared to QPFS on all the datasets. For QPFS-SMO, the savings come from the fact that unlike QPFS, it does not need to calculate the SVD of the Q matrix.

Table 1: Comparison of average time and memory usage.

Dataset	Time(seconds) QPFS	Time(seconds) QPFS-SMO	Memory(KB) QPFS	Memory(KB) QPFS-SMO
Colon	118.0	**4.8**	84779	**16981**
SRBCT	178.7	**6.6**	100591	**89884**
GCM	-	**483.5**	-	**510949**

Accuracy Figure 1 compares the accuracies of the two approaches as we vary the number of features to be selected from 1 to 400. As expected, the accuracies achieved by the two algorithms are quite similar at varying number of top features selected. The error rates come down as relevant features are added to the set. Once the relevant set has been added, any more additional (irrelevant) features lead to loss in accuracy. Results for the other datasets show a similar trend.

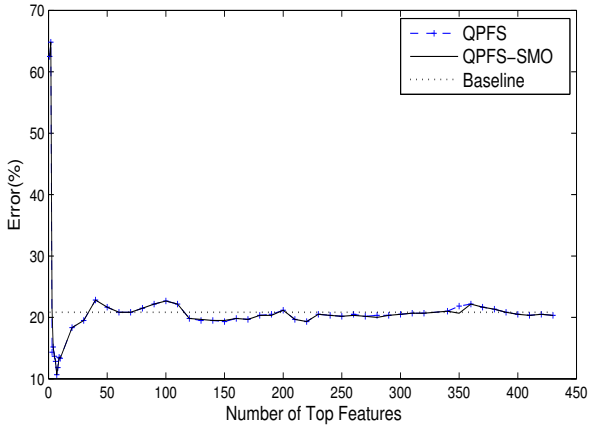

Figure 1: Error rate for Colon Dataset

Conclusion

We have presented an SMO based optimization for QPFS which is significantly more efficient both in time and memory compared to the standard formulation. Directions for future work include experimenting on more datasets, on the fly computation of the similarity matrix and parallel formalism of our SMO based framework.

References

Bekkerman, R.; Yaniv, R. E.; Tishby, N.; and Winter, Y. 2003. Distributional word clusters vs. words for text categorization. *Journal of Machine Learning Research* 3:1183–1208.

Biesiada, J.; Duch, W.; and Duch, G. 2005. Feature selection for high-dimensional data: A kolmogorov-smirnov correlation-based filter. In *In Proc. of CORES 2005, the 4th International Conference on Computer Recognition Systems (2005)*.

Breiman, L., et al. 1984. *Classification and Regression Trees*. New York: Chapman & Hall.

Duda, R. O.; Hart, P. E.; and Stork, D. G. 2000. *Pattern Classification*. John Wiley and Sons, 2nd edition.

Fan, R.-E.; Chang, K.-W.; Hsieh, C.-J.; Wang, X.-R.; and Lin, C.-J. 2008. Liblinear: A library for large linear classification. *Journal of Machine Learning Research* 9:1871–1874.

Fan, R.-E.; Chen, P.-H.; and Lin, C.-J. 2005. Working set selection using second order information for training support vector machines. *Journal of Machine Learning Research* 6:1889–1918.

Forman, G.; Guyon, I.; and Elisseeff, A. 2003. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research* 3:1289–1305.

Forman, G. 2008. BNS feature scaling: an improved representation over tf-idf for svm text classification. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge mining*, 263–270. New York, NY, USA: ACM.

Fowlkes, C.; Belongie, S.; and Malik, J. 2001. Efficient spatiotemporal grouping using the nystrm method. In *In Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 231–238.

Ganesh Kumar, P.; Aruldoss Albert Victoire, T.; Renukadevi, P.; and Devaraj, D. 2012. Design of fuzzy expert system for microarray data classification using a novel genetic swarm algorithm. *Expert Syst. Appl.* 39(2):1811–1821.

Golub, T. R.; Slonim, D. K.; Tamayo, P.; Huard, C.; Gaasenbeek, M.; Mesirov, J. P.; Coller, H.; Loh, M. L.; Downing, J. R.; Caligiuri, M. A.; and et al. 1999. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* 286(5439):531–537.

Guyon, I., and Elisseeff, A. 2003. An introduction to variable and feature selection. *Journal of Machine Learning Research* 3:1157–1182.

John, G. H.; Kohavi, R.; and Pfleger, K. 1994. Irrelevant Features and the Subset Selection Problem. In Cohen, W. W., and Hirsh, H., eds., *Machine Learning, Proceedings of the Eleventh International Conference*, 121–129. Rutgers University, New Brunswick, NJ, USA: Morgan Kaufmann.

Keerthi, S. S.; Shevade, S. K.; Bhattacharyya, C.; and Murthy, K. R. K. 2001. Improvements to platt's smo algorithm for svm classifier design. *Neural Comput.* 13:637–649.

Khan, J.; Wei, J. S.; Ringner, M.; Saal, L. H.; Ladanyi, M.; Westermann, F.; Berthold, F.; Schwab, M.; Antonescu, C. R.; Peterson, C.; and Meltzer, P. S. 2001. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine* 7(6):673–679.

Kohavi, R., and John, G. H. 1997. Wrappers for feature subset selection. *Artificial Intelligence* 97(1):273–324.

Ladicky, L., and Torr, P. 2011. Locally linear support vector machines. In Getoor, L., and Scheffer, T., eds., *Proceedings of the 28th International Conference on Machine Learning*, ICML '11, 985–992. New York, NY, USA: ACM.

Langley, P. 1994. Selection of relevant features in machine learning. In *In Proceedings of the AAAI Fall symposium on relevance*, 140–144. AAAI Press.

Modi, J. 1988. *Parallel Algorithms and Matrix Computation*. Oxford: Oxford University Press.

Peng, H.; Long, F.; and Ding, C. 2005. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27:1226–1238.

Prasad, Y.; Biswas, K. K.; and Jain, C. K. 2010. Svm classifier based feature selection using ga, aco and pso for sirna design. In Tan, Y.; Shi, Y.; and Tan, K. C., eds., *ICSI (2)*, volume 6146 of *Lecture Notes in Computer Science*, 307–314. Springer.

Prasad, Y.; Biswas, K.; and Singla, P. 2013. Scaling-up quadratic programming feature selection: Supplementary material. http://www.cse.iitd.ac.in/∼parags/papers/qpfs-smo-supplement-aaai13.pdf.

Rodriguez-Lujan, I.; Huerta, R.; Elkan, C.; and Cruz, C. S. 2010. Quadratic programming feature selection. *Journal of Machine Learning Research* 11:1491–1516.

Simon, H. U. 2007. On the complexity of working set selection. *Theor. Comput. Sci.* 382:262–279.

Weston, J.; Mukherjee, S.; Chapelle, O.; Pontil, M.; and Vapnik, V. 2001. Feature selection for SVMs. In *Advances in Neural Information Processing Systems (NIPS 13)*, volume 13, 668–674.

Wong, T.-T., and Hsu, C.-H. 2008. Two-stage classification methods for microarray data. *Expert Syst. Appl.* 34:375–383.

Zhang, Y.; Ding, C.; and Li, T. 2008. Gene selection algorithm by combining reliefF and mRMR. *BMC Genomics* 9(Suppl 2):S27.