Evaluating Feature Selection for SVMs in High Dimensions

Roland Nilsson[1], José M. Peña[1], Johan Björkegren[2], and Jesper Tegnér[1]

[1] IFM Computational Biology, Linköping University, SE58183 Linköping, Sweden,
{rolle, jmp, jespert}@ifm.liu.se
[2] Gustav V Research Institute, Karolinska Institute, SE17177 Stockholm, Sweden
johan.bjorkegren@ki.se

Abstract. We perform a systematic evaluation of feature selection (FS) methods for support vector machines (SVMs) using simulated high-dimensional data (up to 5000 dimensions). Several findings previously reported at low dimensions do not apply in high dimensions. For example, none of the FS methods investigated improved SVM accuracy, indicating that the SVM built-in regularization is sufficient. These results were also validated using microarray data. Moreover, all FS methods tend to discard many relevant features. This is a problem for applications such as microarray data analysis, where identifying all biologically important features is a major objective.

1 Introduction

In pattern recognition, feature selection (FS) is traditionally viewed as a preprocessing step that simplifies the task of learning classifiers, rather than a learning objective in itself [1]. On the other hand, there is currently considerable interest within the bioinformatics community to apply FS methods to discover biologically important genes (features) that are not captured by traditional statistical testing [2]. For example, in cancer research, the primary interest is to identify all cancer-related genes from microarray data [3]. This is a fundamentally different problem, because many of the biologically important features may not be needed for classification [4], and will therefore be discarded by FS methods that optimize for classification accuracy.

To assess the applicability of FS methods to microarray data, we performed a systematic evaluation of a number of FS methods in conjunction with SVMs. To our knowledge, this study is the first systematic evaluation of feature set accuracy, and the first to simulate high-dimensional data of the order found in microarray studies.

2 Methods

Throughout, we assume that examples $(x^{(i)}, y^{(i)})$ are independent observations of the random variable pair (X, Y) with distribution $f(x, y)$ on the domain $\mathcal{X} \times \mathcal{Y}$.

J. Fürnkranz, T. Scheffer, and M. Spiliopoulou (Eds.): ECML 2006, LNAI 4212, pp. 719–726, 2006.

We will denote components of a vector x by x_i, and restrictions to a given feature set S by $x_S = \{x_i : i \in S\}$. A classifier is defined simply as a function $g(x) : \mathcal{X} \mapsto \mathcal{Y} = \{-1, +1\}$, predicting a category y for each observed example x. For a given sample size l, a classifier is induced from data $D^l = \{(x^{(i)}, y^{(i)})\}_{i=1}^l \in (\mathcal{X} \times \mathcal{Y})^l$ by an inducer (a learning algorithm), defined as a function $I : (\mathcal{X} \times \mathcal{Y})^l \mapsto \mathcal{G}$, where $\mathcal{G}$ is some set of possible classifiers. The optimal (Bayes) classifier g^* is defined as the one that minimizes the *risk* functional

$$R(g) = \sum_{y \in \mathcal{Y}} p(y) \int_{\mathcal{X}} 1\{g(x) \neq y)\} f(x|y) dx \ , \tag{1}$$

In our simulations we use gaussian densities, for which $R(g)$ is easy to calculate directly from (1), and g^* is unique and can be derived analytically. For comparing learning algorithms, are we interested in the overall performance of an inducer I, rather than the performance of a particular g [5]. We evaluate the performance of I using the *expected* risk

$$\rho = E_D[R(I(D^l))] \ . \tag{2}$$

We will also evaluate accuracy with respect to the set of relevant features. This set is defined as [4]

$$S_{\text{REL}} = \{i : \exists S : p(y|x_i, x_S) \neq p(y|x_S)\} \ , \tag{3}$$

where $p(y|x_S)$ is the conditional density of Y after observing $X_S = x_S$. Informally, a feature is relevant if it carries "information" about the target variable Y. Relevant features are either *strongly* or *weakly* relevant; the latter may be "redundant" in the sense that they are not required for optimal classification [4]. The optimal feature set with respect to classification accuracy is defined as

$$S_{\text{OPT}} = \arg\min_S E_{D_S}[R(I_S(D_S^l))] \ , \tag{4}$$

where I_S is a suitable inducer for the data D_S^l, using the features S. In general, S_{OPT} may not be unique, and the minimization may require an exhaustive search among all subsets of S_{REL}, which is an NP-complete problem [6]. However, in our simulations S_{OPT} was identical to the features used by the Bayes rule g^*. Therefore, we are able to measure feature set precision and recall against both S_{REL} and S_{OPT}. In analogue with the risk measure above, for describing the performance of a FS algorithm we consider the distribution of these measures when D^l is a random variable, and estimate the expected value of this distribution.

An overview of our simulation/evaluation procedure is shown in figure 1. For a given data distribution $f(x, y)$, we first take a sample D^l to be used as training data (step 1). We then perform FS and classifier (SVM) induction. Finally, we calculate classifier error probabilities (steps 3,4), the feature sets S_{REL} and S_{OPT}(step 5) and the precision and recall (step 6) with respect to these sets. For each FS algorithm, this process is repeated 100 times and averaged values are reported.

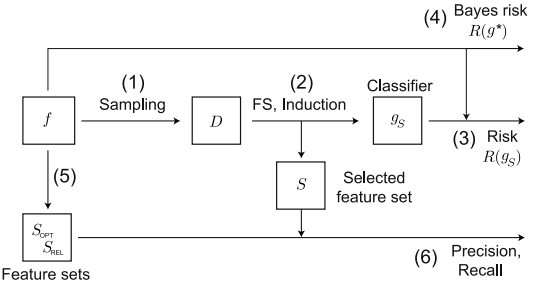

Fig. 1. A schematic view of the evaluation process

We chose five well-known FS methods for evaluation: Pearson Correlation (PC) [1], SVM Naive Weight Rank (WR) [7], Recursive Feature Elimination (RFE) [7], Linear Programming-SVM (LPSVM) [8] and Approximation of the zeRO-norm Minimization (AROM) [9]. PC is a filter method, WR and RFE are wrapper methods, while AROM and LPSVM are embedded methods. We also refer PC, WR and RFE as "ranking methods" since they merely output a ranking of features, while LPSVM and AROM output a set S, thus determining $|S|$ automatically. Throughout, we used a linear SVM [10] as the inducer $I(D^l)$.

For more detailed method descriptions, we refer to the extended version of this paper, available at `www.ifm.liu.se/~rolle/ecml2006.pdf`.

3 Results

We used a gaussian data distribution for all simulations. This was designed so that a subset of m features $X_1, \ldots, X_m$ were relevant to Y, while $X_{m+1}, \ldots, X_n$ were irrelevant. Of the m relevant features, $m/2$ were in the optimal feature set; further, half these $(m/4)$ were *marginally* independent of Y and thus undetectable by univariate filter methods like PC. We sampled 100 training data points and normalized data to zero mean and unit variance before applying each method. The key parameters to the "difficulty" of the learning problems represented by this data distribution are m and n. We chose a parameter grid $8 \leq m \leq 500$, $20 \leq n \leq 5000$, with values evenly spaced on a logarithmic scale (figure 2A).

The expected risk ρ for the SVM without FS on the (m, n) parameter grid is shown in figure 2A. We find that ρ increases slightly with n, but decreases rapidly with respect to m. Thus, more features is in general better for the SVM: as long as we can obtain a few more relevant features, we can afford to include many irrelevant ones. Therefore, improving SVM performance by FS is very difficult. In particular, an FS method must provide very good recall, or SVM performance will degrade quickly.

To validate our results, we also tested the FS methods on a large microarray data set consisting of 12,600 features (genes) and 136 samples [11]. For

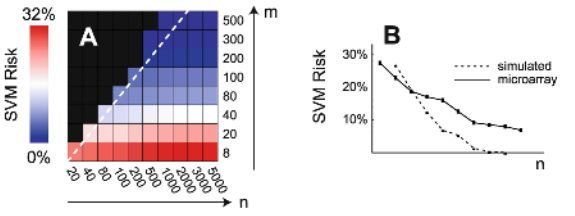

Fig. 2. A: Plot of expected SVM risk ρ *vs.* number of relevant features m and total number of features n. **B:** Dotted line, plot of SVM risk *vs.* n for simulations, corresponding to the dotted diagonal in (A). Solid line, SVM risk *vs.* n for microarray data. Here m is unknown but proportional to n.

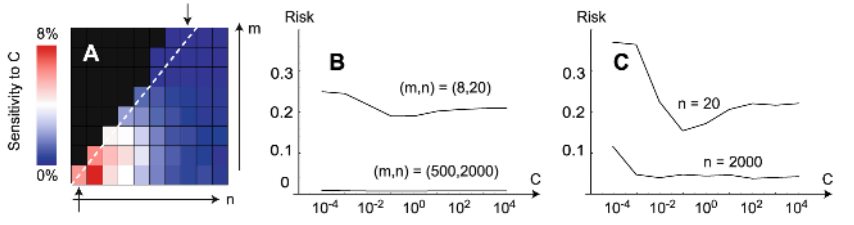

Fig. 3. Sensitivity to the SVM C-parameter. **A:** Sensitivity defined as $\max_C \hat{R} - \min_C \hat{R}$ plotted against m and n. **B:** For simulated data, detailed plot of $\hat{R}$ against C for the cases (m, n) marked by arrows in (A), roughly corresponding to the cases in (C). **C:** For microarray data, plot of $\hat{R}$ against C for $n = 20$ and $n = 2000$.

comparison with our simulations, we first extracted the 5000 features with largest variance and then extracted random subsets of sizes $10, \ldots, 5000$ from these. Although m is unknown in this case, in expectation this procedure gives a constant m/n ratio, since we draw features with equal probability. This roughly corresponds to a diagonal in figure 2A. We selected random training sets of $l = 100$ samples, estimated $\hat{R}(g)$ on the remaining 36 samples, and repeated this process 300 times for each n. The resulting risk estimate was found to agree qualitatively with our simulations (figure 2B).

The value of the regularization parameter C has been found to strongly impact SVM classification accuracy in low dimensions [12]. In our simulations, we optimized C over a range $10^{-4}, \ldots, 10^4$ for each (m, n). We found that C was no longer important in higher dimensions (figure 3A), regardless of the value of m. At lower dimensions, $C \approx 1$ provided good performance (figure 3B), so we fixed $C = 1$ for the remainder of this study. We observed the same (and even stronger) trend for the microarray data (figure 3C). We conclude that the parameter C has virtually no impact on classification accuracy in high dimensions.

Next, we investigated the accuracy of the feature rankings produced by PC, WR and RFE. To address this question without involving the problem of choosing $|S|$, we constructed ROC-curves (figure 4). We found that RFE outperforms

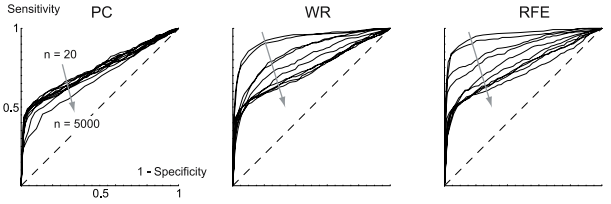

Fig. 4. ROC-curves for the PC, WR and RFE methods. Here we fixed $m = 8$ relevant features and varied $n = 20, \ldots, 5000$ as indicated by grey arrows. Dashed diagonals indicate expected result for randomly selected features.

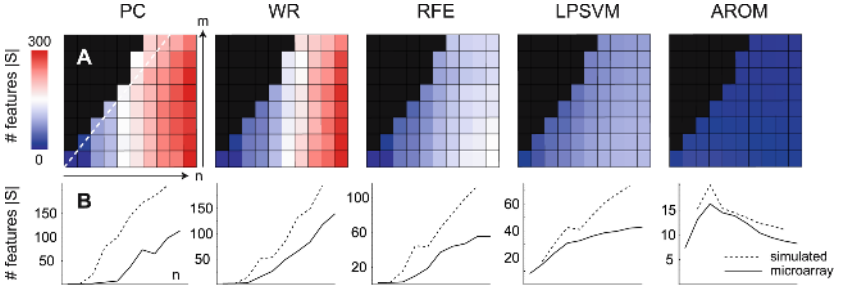

Fig. 5. A: Number of selected features $|S|$ for each (m, n) for simulated data. All plots are scaled equally to (0,300). **B:** Number of selected features $|S|$ *vs.* n, corresponding to the dashed diagonal in (A), for simulated and microarray data. Plots are scaled differently.

WR, which in turn outperforms PC, in agreement with Guyon et al. [7]. This was expected, since 1/4 of the relevant features are not detectable by PC. However, these differences diminished with increasing n. At $n = 5000$, the simpler WR method was as accurate as RFE.

To use ranking methods in practise, a critical issue is how to determine $|S|$ (LPSVM and AROM decide this automatically by heuristics that favor small feature sets [8,9]). A common strategy is to minimize some risk estimate $\hat{R}(g_S)$ for the classifier g_S induced using the feature set S, over a number of possible choices of $|S|$ [13]. For this purpose, we chose the radius-margin bound [14, section 10.7]. Overall, we found that the ranking methods tend to select more features than AROM or LPSVM (figure 5A). We also found that $|S|$ tends to *increase* with n. This can be explained by noting that with better rankings we reach low classifier risk $R(g_S)$ for small $|S|$. Therefore, PC chooses the largest $|S|$, and RFE the smallest. This was also verified for the microarray data (figure 5B). More surprisingly, there was also a tendency for $|S|$ to *decrease with* m (most evident for RFE). This can be understood in the same fashion: the FS problem becomes harder as m decreases, so rankings become more inaccurate and a larger $|S|$ must be used. We conclude that, by selecting $|S|$ to minimize

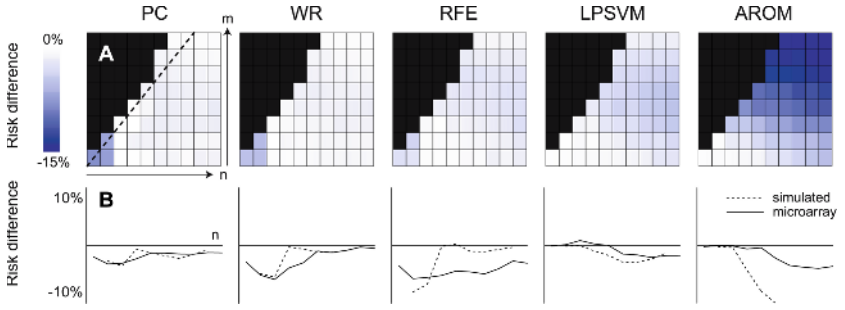

Fig. 6. A: Risk difference $\rho(g) - \rho(g_S)$, using each respective FS method to obtain S (negative means worse accuracy with FS), simulated data. **B:** Estimated risk *vs.* n, corresponding to the dashed diagonal in (A), for simulated and microarray data.

risk, we obtain methods that attempt to control recall but sacrifice precision. LPSVM produced smaller feature sets than RFE, but otherwise exhibited the same tendencies discussed above. Again, the simulation results were consistent with microarray data (figure 5B).

In principle, if $\hat{R}(g_S)$ is accurate, then optimizing this estimate over $|S|$ should at least guarantee that ranking methods do not increase classifier risk. Our simulations verified this: in figure 6A, the difference $\rho(g) - \rho(g_S)$ is close to 0. Thus, the radius-margin bound seems to be accurate, so our results should be attributed to the rankings themselves. LPSVM and AROM worked best around $n \approx 100$ (figure 6A,B), corresponding to the data sets used in the original publications [8,9]. In higher dimensions however, these methods tend to increase the SVM risk. AROM in particular increased ρ by up to 15%, probably because it insisted on very small feature sets. Results on microarray data were similar (figure 6B) except possibly for RFE, which was less accurate on microarray data. In summary, none of the FS methods improved SVM accuracy.

For the simulated data, we measured the accuracy of selected feature sets by precision and recall *vs.* S_{OPT} (figure 7A,B) and S_{REL} (figure 7C,D). There are interesting differences between these two feature sets. Concerning recall *vs.* S_{REL}, we see that PC > WR > RFE > LPSVM > AROM. The filter method PC presumably performs best here since it does not distinguish between strong and weak relevance. In fact, we see that PC selects more weakly than strongly relevant features (since it gives lower recall *vs.* S_{OPT}). In contrast, RFE, LPSVM and AROM have higher recall *vs.* S_{OPT} than *vs.* S_{REL}. All of these methods involve some form of risk optimization, and therefore target S_{OPT}. Consequently, they tend to miss (or, avoid, depending on one's perspective) many of the weakly relevant features. All methods have low precision; AROM provided the best precision, but at the price of lower recall. This is natural since AROM was biased towards very small $|S|$ (figure 5). The remaining methods were comparable.

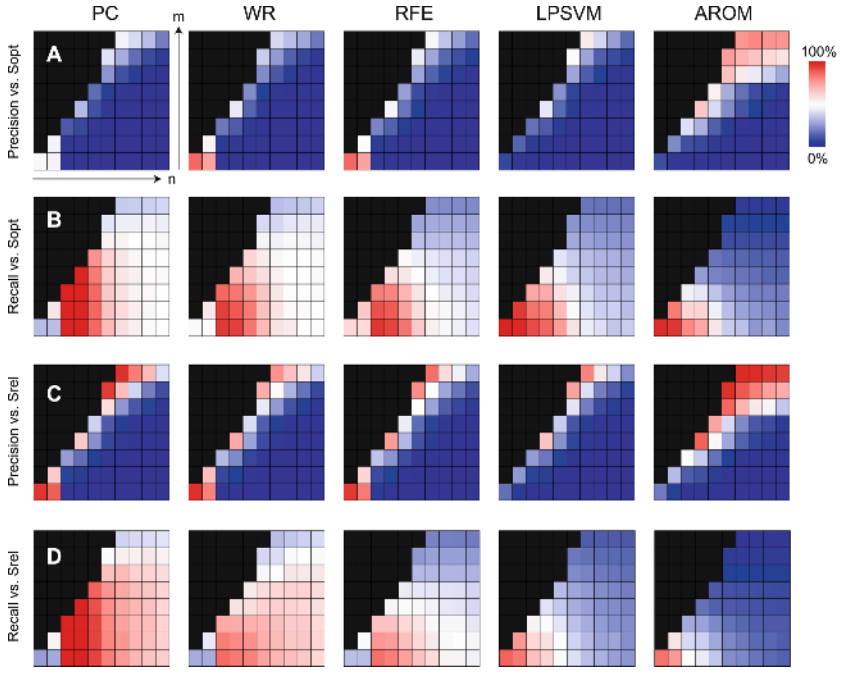

Fig. 7. Feature set accuracy measures for each FS method

4 Discussion

A striking trend in our results is that both classification and feature selection (FS) methods behave very differently in high *vs.* low dimensions. For example, while AROM and LPSVM improve classification accuracy for SVMs for lower n and $m \ll n$ [8,9], we find no improvement at high n (figure 6). Also, while the C-parameter is crucial for low n, it has little or no influence at high n (figure 3). Thus, we recommend that simulation studies of FS methods are performed with dimension comparable to that of real data.

None of the FS methods tested improved SVM classification accuracy in high dimensions. To explain this, one may consider FS as a minimization of the L_0-norm of a vector of feature weights, while the SVM minimizes the L_2-norm [15]. Our results the imply that in high dimensions, the L_2-norm is simply the better choice. For multi-class problems however, there are indications that FS may improve SVM performance [16].

In microarray data analysis, it is often desirable to control precision while maximizing recall [17]. It is clear from figure 7 that none of the methods tested provide such control. A feature selection method that solves this problems would be most useful for microarray data analysis.

Acknowledgements

This work was supported by grants from the Ph.D. Programme in Medical Bioinformatics, the Swedish Research Council (VR-621-2005-4202), Clinical Gene Networks AB and Linköping University.

References

1. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research **3** (2003) 1157–1182
2. Dougherty, E.R.: The fundamental role of pattern recognition for the gene-expression/microarray data in bioinformatics. Pattern Recognition **38** (2005) 2226–2228 Editorial.
3. Golub, T.R., et al.: Molecular classifiation of cancer: class discovery and class prediction by gene expression monitoring. Science **286** (1999) 531–537
4. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence **97** (1997) 273–324
5. Dietterich, T.G.: Approximate statistical tests for comparing supervised classification learning algorithms. Neural Computation **10** (1998) 1895–1923
6. Davies, S., Russel, S.: NP-completeness of searches for smallest possible feature sets. In: Proceedings of the 1994 AAAI fall symposium on relevance, AAAI Press (1994) 37–39
7. Guyon, I., et al.: Gene selection for cancer classification using support vector machines. Machine Learning **46** (2002) 389–422
8. Fung, G., Mangasarian, O.L.: A feature selection newton method for support vector machine classification. Computational Optimization and Applications **28** (2004) 185–202
9. Weston, J., et al.: Use of the zero-norm with linear models and kernel methods. Journal of Machine Learning Research **3** (2003) 1439–1461
10. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning **20**(3) (1995) 273–297
11. Singh, D., et al.: Gene expression correlates of clinical prostate cancer behavior. Cancer Cell **1** (2002) 203–209
12. Keerthi, S.S.: Efficient tuning of SVM hyperparameters using radius/margin bound and iterative algorithms. IEEE Transactions on Neural Networks **13**(5) (2002) 1225–1229
13. Ambroise, C., McLachlan, G.J.: Selection bias in gene extraction on the basis of microarray gene-expression data. PNAS **99**(10) (2002) 6562–6566
14. Vapnik, V.N.: Statistical Learning Theory. John Wiley and Sons, Inc. (1998)
15. Perkins, S., et al.: Grafting: Fast, incremental feature selection by gradient descent in function space. Journal of Machine Learning Research **3** (2003) 1333–1356
16. Statnikov, A. et al.: A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. Bioinformatics **21**(5) (2005), 631–643
17. Speed, T. (ed.): Statistical Analysis of Gene Expression Microarray Data. Chapman & Hall (2003)