Stepwise Feature Selection Using Multiple Kernel Learning

Vilen Jumutc

Riga Technical University, Meza 1/4, LV-1658 Riga, Latvia
Jumutc@gmail.com

Abstract. In this paper we propose a novel more flexible approach for the simultaneous feature selection and classification using Support Vector Machine and recent major advances of it, namely Multiple Kernel Learning. Using a quite simple kernel assembly scheme in the following paper we will indicate that feature selection and classification could be done in one step without applying computationally intensive and maybe inadequate filtering or wrapper approach. Later imply that to achieve dimensionality reduction, tractable and more compact as well as comprehensively accurate model it is necessary to accomplish all of above goals by "training" SVM only once. Actually we apply some additional prerequirement that resulted in a ranking criteria that could be provided by any domain expert or created by our algorithm using Linear SVM by itself. Provided experimental results verify that our approach is comparable or even more accurate and robust than other feature extraction/selection schemes tested on public UCI datasets.

1 Introduction

Recent advances in computer science and computational intelligence uncover vital necessity for the feature selection and dimensionality reduction methods applied to the variety of highly-dimensional data sources like biomedical CT images, cardiogram, microarray and other data with high variance and insufficient sample size. By this research we intend to resolve simultaneously several problems of previous feature selection/extraction methods like SVM-RFE [1] that solely depends on Linear SVM and like every wrapper approach evaluates classifier each iteration of feature extraction algorithm. We state that our feature selection scheme is both computationally inexpensive and outperforms resembling approaches that basically implement either forward-selection procedure to ensure crisp feature selection or backward-elimination that potentially could be very time-consuming and suffers from overall non-convexity of stated optimization problem. Embedded MKL extension provides us with strong convexity of feature selection problem and simultaneously helps to build ad-hoc classifier that incorporates only most predictive and discriminative attributes.

The upcoming sections of our paper are structured as follows: Section 2 briefly presents common SVM basics and MKL extension. Section 3 describes in details our feature selection method and presents generalized algorithm. Section

4 summarizes experimental setup and numerical results. And finally in Section 5 we analyze and compare our method with other feature extraction/selection approaches as well as conclude about further possible research area.

2 Background

In this section we present some commonly recognized SVM basics [2] and MKL extension of it [4, 5] for learning from an affine combination of regular (linear, RBF, polynomial etc.) or data-driven kernels.

2.1 Support Vector Machine

Support Vector Machine is based on the concept of separating hyperplanes that define decision boundaries using Statistical Learning Theory [2]. Using a kernel function, SVM is an alternative training method for polynomial, RBF and multi-layer perceptron classifiers in which the optimal solution or decision surface is found by solving the quadratic programming problem with linear constraints, rather than by solving a non- convex, unconstrained minimization problem as stated in typical back-propagation neural network.

Further we present only dual representation of SVM primal objective that is expressed in terms of its Lagrangian multipliers λ_i and can be effectively optimized using any off-shell linear optimizer that supports constraint adaptation:

$$\max_{\lambda}\{\sum_{i=1}^{l}\lambda_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\lambda_i\lambda_i y_i y_j K(x_i, x_j)\}, \quad \lambda_i \geq 0, \sum_{i=1}^{l}\lambda_i y_i = 0, \tag{1}$$

where λ_i represents a Lagrangian multiplier, y_i is $\{\pm 1\}$ - valued label of data sample x_i, $K(x_i, x_j)$ is a kernel function and l is a number of training samples.

Finally corresponding classification of a new sample x' is derived by: $d = sign(\sum_i \lambda_i K(x_i, x') + b)$, where $K(x_i, x')$ and b correspond to a kernel function evaluated for a new sample and a linear offset of the optimal decision hyperplane.

2.2 Multiple Kernel Learning

Multiple Kernel Learning aims at simultaneously learning the kernel and the associated predictor in general SVM context. Recent applications of MKL have clearly proven that using multiple kernels instead of a single one can enhance the interpretability of the decision function and improve performances [4, 5]. In such cases, a convenient approach is to consider that the kernel $K(x, y)$ is actually a convex combination of basis kernels:

$$K(x, y) = \sum_{i=1}^{m} w_i K_i(x, y), \quad w_i \geq 0, \sum_{i=1}^{m} w_i = 1, \tag{2}$$

where m is the total number of kernels. Within this framework, the problem of data representation through the different kernels is then transferred to the choice of optimal weights w_i that minimizes the MKL objective function [5].

3 Proposed method

In this section we describe in details aforementioned feature selection method and a general kernel assembly scheme for Multiple Kernel Learning. The overall approach is given in the form of abstract algorithm that depicts a clear view of all steps needed to implement proposed method.

3.1 Ranking criteria

Before handling actual feature selection procedure we apply some additional ranking criteria that performs an ordering of all features according to their relevant importance to an evaluated classifier. Similar approach was provided by [1] in SVM-RFE method and consists of the following very simple steps:

1. Evaluate Linear Support Vector Machine and compute corresponding weight vector of dimension length: $w = \sum_i \lambda_i y_i x_i$, where λ_i is a dual variable of SVM optimization problem, y_i is a label of *i-th* training sample x_i
2. Compute the ranking $c_j = (w_j)^2$ for every *j-th* attribute
3. Sort the ranked attributes in the descending order and create corresponding ordered list of features S

3.2 Generalized algorithm

After evaluating the ranking criteria and obtaining ordered list of features we perform following kernel assembly scheme that could be effectively summarized by the generalized algorithm that incorporates several subroutines and inner algorithms such that *SimpleMKL* [3], *InitKernelMatrices* etc.:

Algorithm 1: Stepwise feature selection via kernel assembly scheme

input : ordered list of features S of size m, training data X of size $n \times m$, class labels Y of size n

output: nonlinear SVM model: λ defines a dual SVM solution and b corresponds to a linear offset, selected feature subsets which correspond to not-null elements of weight vector w

```
1  begin
2      K ← InitKernelMatrices ();
3      I_RBF ← InitRBFInterval ();
4      for i ← 1 to m do
5          S' ← S(1, i);
6          X' ← X(:, S');
7          for j ← 1 to |I_RBF| do
8              ind ← (i − 1) × |I_RBF| + j;
9              K[ind] ← ComputeRBFKernel (X', I_RBF[j]);
10         end
11     end
12     [w, λ, b] ← SimpleMKL (Y, K);
13 end
```

Finally classification using defined in Algorithm 1 SVM model could be handled using following equation:

$$d = sign(\sum_i \sum_j \lambda_i w_j K_j(x_i, x') + b), \tag{3}$$

where K_j is the RBF kernel function and x' is a test sample.

It is obvious that represented by Algorithm 1 kernel assembly scheme could be summarized as a stepwise feature subset selection from the ordered list of all attributes. Further algorithmic steps only broaden number of kernel matrices by additional parametrization of RBF kernel.

To implemented our approach we have selected to train and test our method within SimpleMKL framework [3] in order to avoid time-expensive cross-validation and provide more accurate estimation of "tuning" parameters of RBF kernel. The later parameters are defined by *InitRBFInterval* method of our generalized algorithm and correspond to unknown optimal bandwidth γ of any RBF kernel.

To fasten computation of incredibly many kernel matrices (in Algorithm 1 number of kernel matrices is bounded by $m \times |I_{RBF}|$) we have decided to estimate optimal iteration pace of the outer "for" loop in our generalized algorithm. In order to lower a computational effort and memory load without significant performance degradation we conducted 10-fold cross-validation on the training set and averaged total error across all folds. The pace with the lowest averaged error was selected for performing Algorithm 1.

4 Experiments

4.1 Experimental Setup

In our experiments we have tested proposed model under predefined $C = 10$ (error trade-off) value of the soft-margin SVM that showed most comprehensible performance for imbalanced data sets and varying γ value of RBF kernel that trade-offs kernel smoothness and could be effectively estimated via SimpleMKL framework [3].

To verify and test our proposed approach we have selected several highly dimensional public UCI datasets and evaluated them under following experimental setup: for datasets that weren't originally separated into validation and training sets we performed 10-fold cross-validation and collected averaged total error and balanced error rate (BER). For others we tested our approach on presented in UCI repository validation set and collected single total error and BER. Additionally we experiment with highly dimensional gene microarray dataset, namely CNS-ET, that was very comprehensively inspected in [6]. For this dataset we apply Leave-One-Out cross-validation scheme to provide comparable results with [6] where Pomeroy et al. followed the same experimental setup.

4.2 Numerical Results

In the following subsection we have summarized numerical results for all datasets under fixed C parameter and enclosed subspace for γ parameter of RBF kernel with some initial guess of its corresponding scaling factor[1]. In the Table 1 we present performance measures obtained by our approach under SimpleMKL framework, linear/nonlinear SVM benchmark results as well as some additional performance measures for SVM with differently applied filtering or wrapper feature selection approach. In braces we give averaged number of selected features for all presented in Table 1 approaches except linear/nonlinear SVM that was "trained" using all features.

Table 1. Averaged Total Error/BER

Dataset	SVM_{linear}	SVM_{rbf}	Our method	F+SVM[a]	CSA[b]
Arrythmia	0.26/0.26	0.25/25	0.21/0.22(34)	-/-	0.26/-(28)
Arcene	0.17/0.18	0.2/0.22	0.13/0.14(1101)	-/0.21(661)	0.19/-(600)
Dexter	0.07/0.07	0.11/0.11	0.08/0.08(118)	-/0.08(209)	0.07/-(717)
CNS-ET	0.33/0.4	0.35/0.5	0.2/0.24(132)	-/-	-/-

[a] SVM with the F-score feature selection scheme [7]
[b] Contribution-Selection Algorithm with the best performing inducted classifier [8]

5 Results Analysis and Conclusion

In this paper we propose novel stepwise feature selection method that basically extends Multiple Kernel Learning approach and helps to provide classifier with the most comprehensible and meaningful subset of features and perform actual classification all in one step. As we can see from the above given experimental results our feature selection method is comparable or even more accurate and robust than other feature selection/extraction approaches. Remarkably that our approach almost anywhere attains comparable or even smaller subset of features. For UCI datasets it is clear that our stepwise feature selection algorithm brings necessary discrimination capabilities and additional accuracy to the nonlinear SVM classifier eliminating noisy and redundant features. Separately we should examine CNS-ET dataset because we do not provide performance measures for F-SVM and CSA methods. In original work of Pomeroy et al. [6] authors preselected 150 most discriminative genes and conducted SVM classifier. They

[1] We have defined range of $b_\gamma \cdot 10^{[-20\ldots20]}$ with the step 0.5 resulting in a total of 81 γ-parameters where b_γ is a corresponding scaling factor of γ stated as follows: $b_\gamma = 1/2 \cdot \sqrt{median(X)}$ where X is a vector of all dataset values.

achieved total error of 25% and balanced error rate (BER) of 29.1%. In comparison we achieve drastically more robust and accurate result without even knowing the domain field. In [6] the best possible result was achieved using the combination of three classifiers (SVM, k-NN and TrkC) and was comparable to our results: total error of 20% and BER of 24.2%. In conclusion we should highlight that our approach is a general purpose algorithm and could be used for any classification problem that suffers from "dimensionality curse" and needs a quick and elegant feature extraction approach for the kernel-based classifiers. Our further research is closely related to the feature ranking algorithms that could definitely provide more reliable and domain-specific information about feature relevance to the problem than ordinary Linear Support Vector Machine.

References

1. Guyon, I. et al.: Gene Selection for Cancer Classification using Support Vector Machines. Machine Learning, Vol.46, Issue 1-3, 389 – 422 (2002)
2. Vapnik, V.: The Nature of Statistical Learning Theory. Springer-Verlag (1995)
3. Rakotomamonjy, A. et al.: SimpleMKL. Journal of Machine Learning Research, Vol.9, 2491–2521 (2008)
4. Lanckriet, G. et al.: Learning the Kernel Matrix with Semidefnite Programming. Journal of Machine Learning Research, Vol.5, 27–72 (2004)
5. Bach, F. et al.: Multiple kernel learning, conic duality, and the SMO algorithm. In Proceedings of the 21st International Conference on Machine Learning. Montreal, Canada, 41–48 (2004)
6. Pomeroy, S. et al.: Prediction of central nervous system embryonal tumour outcome based on gene expression. Nature, Vol.415, 436–442 (2002)
7. Chen, Y-W. and Lin C-J.: Combining SVMs with Various Feature Selection Strategies. Studies in Fuzziness and Soft Computing, Vol.207, 315–324 (2006)
8. Cohen, S. et al.: Feature Selection Based on the Shapley Value. In Proceedings of the 19th International Joint Conference on Artificial Intelligence. Edinburgh, Scotland, UK, 665–670(2005)