

Feature Similarity Based Redundancy Reduction for Gene Selection

X. Fu, F. Tan, H. Wang, Y-Q. Zhang, R. Harrison

Abstract—In this paper we propose a feature similarity based redundancy reduction (FSRR) algorithm for high-dimensional gene expression data analysis. FSRR has two steps. First, the relevance of each feature is evaluated. Second, based on the relevance, the redundant features are removed by feature similarity. The efficiency and effectiveness of our algorithm is established through an experimental study using gene expression data. Four state-of-art feature ranking algorithms and three feature similarity measures are compared and discussed in our work. The results indicate that our algorithm has the capability of finding a well-suited feature set and improving the classification accuracy.

1. INTRODUCTION

Feature selection is an important technique in mining large data set. Obtaining a smaller set of representative features, remaining the characteristics of the data, is very helpful to save processing time and build more accurate learning system.

Many feature selection methods have been proposed [1]. They can be broadly grouped into two categories: filter model and wrapper model. In filter model, features are evaluated based on the general characteristics of the data without relying on any mining algorithms. On the contrary, wrapper model requires one mining algorithm and uses its performance to determine the goodness of feature sets. The wrapper model searches features better suited to the mining algorithm thus achieves higher accuracy. However, it is more computationally expensive than filter model.

The major challenge of gene expression data analysis is the large number of genes compared to the small number of samples in a typical experiment. Only some of the genes are informative while many other genes are redundant [2]. These redundant genes introduce unnecessary noise to the microarray analysis and result in computational difficulties for clustering and classification. Therefore, removing redundant and irrelevant genes and finding a subset of discriminative genes is crucial for gene expression data analysis. Feature ranking methods are widely used for gene selection. However, the implementation involves two problems:

(1) How many genes should be selected? People use top ranked genes heuristically but cannot determine the optimal feature set.

(2) Which genes are enough to describe the pattern of data and which genes are redundant or even noisy?

Top ranked genes may be highly correlated and cannot cover all characteristics of data. Therefore only using top ranked features cannot achieve good performance. In this paper, we suggest a two-step algorithm, named as FSRR, for feature selection by taking feature relevance and redundancy into consideration. The features have been ranked in the non-increasing order of relevance. FSRR starts with the top feature (the most relevant feature) and filter features one by one in order based on the feature similarity between a feature and the selected feature set. Therefore FSRR has the ability to select both top ranked features and low ranked features. We test our algorithm using four feature ranking algorithms (t-test[2], fisher[3], SVM-RFE[4] and AROM-SVM[5]) and three feature similarity measures (Correlation coefficient, Least square regression error, and Maximal information compression index [6]) with Prostate cancer data[7].

The rest of the paper is organized as follows. In Section 2, we introduce the feature ranking and feature similarity and present our algorithm. Experiment results are shown in Section 3. Conclusion is drawn in Section 4.

2. ALGORITHM

2.1. Feature ranking and feature similarity

Feature selection methods search for the best feature set in order to reduce dimensionality and improve the classification accuracy. Exhaustive search on all possible subsets of features can reveal the optimal feature set but is too time-consuming, sometimes unfeasible for larger number of features. Therefore, heuristic search is necessary. To carry out heuristic search, we need to know which features are relevant and which are redundant. Feature ranking methods generally evaluate the relevance of each feature using a scoring function and sort the features in the decreasing order of relevance. Feature ranking reveals the relevance of features but give no information about the redundancy. Redundancy can be observed by measuring the similarity between features. We make use of linear dependency as the similarity measure in our work. For a linearly separable data, it is known that if feature set A has linear dependency with feature set B, the data is still linearly separable if feature set A is removed [8]. Using both feature ranking and feature similarity, we present a feature selection algorithm for better classification.

2.2. FSRR

This work is supported in part by NIH under P20 GM065762.

X. Fu, F. Tan, H. Wang, Y-Q. Zhang, and R. Harrison are with the Department of Computer Science, Georgia State University, Atlanta, GA, 30303, USA. Fax: (404) 463-9912 (e-mail: xfu1@gsu.edu, ftan@gsu.edu, hwang10@gsu.edu, zhang@taichi.cs.gsu.edu, cscrwh@asterix.cs.gsu.edu)

Given a set of features ranked in non-increasing order of relevance by a feature ranking algorithm, FSRR starts with the top feature and filter features one by one in order based on the feature similarity between a feature and the selected feature set. Therefore FSRR is able to select both top ranked features and low ranked features. The pseudo code of our algorithm is shown in Figure 1.

<p>Input: data Output: feature_set</p> <ol style="list-style-type: none"> 1. Let $L=\{f_1, f_2, \dots, f_n\}$, n is the number of features. Features in L are sorted in non-increasing order of relevance given by a feature ranking algorithm. 2. Let feature_set=$\{ f_1 \}$ 3. for each $f_i \in L$ and $i \in [2, n]$ 4. sum=0; 5. siz=length(feature_set); 6. for each $f_j \in$ feature_set and $j \in [1, \text{siz}]$ 7. sum=sum+ <i>Similarity</i>(f_j, f_i) 8. end for 9. if sum/siz < δ 10. feature_set = { feature_set, f_i } 11. end for

Figure 1. Pseudo code of FSRR.

In Figure 1, δ is a constant. *Similarity* is a function computing similarity between the two features. If the average similarity between a feature and the feature_set is less than a certain value, this feature is added into the feature_set, otherwise ignored. The worst case of this algorithm is $O(n^2)$.

3. EXPERIMENTS

In this section, we experimentally evaluate our algorithm by comparing four feature ranking algorithms and three feature similarity measures on Prostate cancer data.

3.1 Data set

We choose the Prostate cancer data from UCI databases [7]. This dataset contains independent training dataset and testing dataset. The training set contains 52 prostate tumor samples and 50 normal prostate samples with around 12600 genes. The testing dataset has 25 tumor and 9 normal samples that are obtained from a different experiment. The data are normalized into the range of [0,1].

3.2 Feature ranking algorithms

In this paper, we study four state-of-art feature ranking algorithms. t-test[2], fisher[3], SVM-RFE[4] and AROM-SVM[5].

3.2.1. Filter-model algorithms

t-test: each sample belongs to one class + or -. For each feature f_i , the mean and standard deviation are calculated using only the samples labeled +. Then a score $T(f_i)$ is given by :

$$T(f_i) = \frac{|\mu_i^+ - \mu_i^-|}{\sqrt{\frac{(\sigma_i^+)^2}{n_+} + \frac{(\sigma_i^-)^2}{n_-}}}$$

Where n_+ is the number of samples labeled as 1.

Those features with the highest scores are considered as the most discriminatory features.

fisher: 'fisher' is a classical measure to assess the degree of separation between two classes. It is similar to t-test. The score function is defined as:

$$T(f_i) = \frac{(\mu_i^+ - \mu_i^-)^2}{(\sigma_i^+)^2 + (\sigma_i^-)^2}$$

3.2.2. Wrapper-model algorithms

SVM-RFE: this method is proposed by Guyon et al.[4] and based on the concept of margin maximization.

The importance of each feature is determined by its influence on the predefined objective function. The objective function J is given by $J = \|w\|^2 / 2$. For feature i , $w_i = (\mu_i(+)-\mu_i(-)) / (\sigma_i(+)+\sigma_i(-))$, where μ_i and σ_i are mean and standard deviation of the feature i for all samples of class(+) and class(-).

SVM-RFE is a backward feature elimination method using w_i^2 as the ranking criterion. Features are removed in an iterative procedure. In each iteration, the SVM is trained and the features that minimize the change of objective function are removed (typically only one feature).

AROM-SVM: Weston et al.[5] suggested minimizing the zero-norm $\|w\|^0 = \text{cardinality}(\{w_j : w_j \neq 0\})$ instead of minimizing the l_1 -norm or l_2 -norm as in standard SVMs. An approximately method was proposed to solve the l_0 -norm formulation of SVMs. The method works in a backward elimination procedure using $|w_i|$ as the ranking criterion.

3.3. Feature similarity measures

We present three similarity measures to reduce redundancy for feature selection.

3.3.1. CC (Correlation coefficient):

$$\rho(x, y) = \frac{\text{cov}(x, y)}{\sqrt{\text{var}(x) \text{var}(y)}}$$

Where x and y are two features. cov() represents the covariance between two features. var() indicates the variance of a feature.

$|\rho(x, y)|$ is used as the similarity measure. $|\rho(x, y)|$ ranges on $[0, 1]$. If two features are linear dependent, $|\rho(x, y)|$ is 1. If two features are completely uncorrelated, $|\rho(x, y)|$ is 0. Larger $|\rho(x, y)|$ value indicates higher dependency.

3.3.2. LSRE (Least square regression error):

The dependency between two features x and y is the mean square error given by

$$e(x, y) = \text{var}(y)(1 - \rho(x, y)^2)$$

$e(x, y)$ ranges on $[0, \text{var}(y)]$. If two features are linear dependent, $e(x, y)$ is 0. If two features are completely uncorrelated, $e(x, y)$ is $\text{var}(y)$. Larger $e(x, y)$ value indicates less dependency.

3.3.3. MICI (Maximal information compression index):

$$2\lambda_2(x, y) = \text{var}(x) + \text{var}(y) - \sqrt{(\text{var}(x) + \text{var}(y))^2 - 4\text{var}(x)\text{var}(y)(1 - \rho(x, y)^2)}$$

λ_2 is the eigenvalue for the direction normal to the principle component direction of features x and y . λ_2 ranges on $[0, 0.5(\text{var}(x) + \text{var}(y))]$. If two features are linear dependent, λ_2 is 0. If two features are completely uncorrelated, λ_2 is $0.5(\text{var}(x) + \text{var}(y))$. Larger λ_2 value indicates less dependency.

3.4. Validation and classifier

We evaluate the various feature ranking methods and similarity measures using Leave-one-out cross-validation (LOO) and testing accuracy. LOO is a technique where a classifier successively learns on $m-1$ samples and tested on the remaining one. This is repeated m iterations and each sample is left out once. Testing accuracy requires the whole data set is partitioned into training set and testing set. The classifier is first trained with training data and then tested with testing data. The two validation measures differ in the choosing training data and testing data. However, how to separate the whole data into training and testing data is an open problem. LOO is known to be a high variance estimator of generalization error [9]. The performance of these two measures is discussed based on our experiments in Section 3.5.

SVM (Support Vector Machine) with linear kernel is used as the classifier. SVM has been compared with other models [4, 10, 11] and is believed to be robust with sparse and noisy data. Our feature selection and classification experiments are completed in the Spider [12] environment running on a PC with Pentium 4(2.64MHz) and 512M RAM.

3.5. Experimental results

3.5.1. LOO accuracy

The LOO accuracy of t-test, ‘fisher’, SVM-RFE, and AROM-SVM is obtained by using top-ranked genes (features). It is shown in Table 1. The highest accuracy of each method is in bold.

Table 1. LOO accuracy on Prostate cancer data

# of Top features	t-test	fisher	SVM-RFE	AROM-SVM
10	92.64	90.44	97.79	100.00
20	88.97	91.17	97.05	100.00
50	91.91	93.38	100.00	100.00
100	95.58	94.11	100.00	100.00
200	95.58	94.11	100.00	100.00
500	92.64	93.38	100.00	100.00
1000	95.58	94.11	100.00	100.00
2000	96.32	96.32	100.00	99.26

From Table 1, we can see that AROM-SVM achieves the best LOO accuracy compared with the others. SVM-RFE gives closer LOO accuracy to AROM-SVM but a little bit lower. The performance of T-test and ‘fisher’ is close to one another but is not as good as SVM-RFE or AROM-SVM. From Table 1, we obtain two observations. First, filter-model algorithms are not as good as wrapper-model algorithms. Second, more features is helpful to achieve higher LOO accuracy if the number of features are confined in a reasonable range. Are these observations correct? More experiments need to be done before drawing a conclusion.

3.5.2. Testing accuracy

We further use the testing accuracy to assess the four feature ranking methods and the results are described in Table 2. The highest accuracy of each method is in bold.

Table 2. Testing accuracy on Prostate cancer data

# of Top features	t-test	fisher	SVM-RFE	AROM-SVM
10	91.17	91.17	85.29	88.23
20	73.52	73.52	97.05	88.23
50	64.70	64.70	97.05	94.11
100	64.70	61.76	97.05	94.11
200	64.70	64.70	91.17	91.17
500	67.64	67.64	94.11	94.11
1000	73.52	70.58	94.11	91.17
2000	73.52	73.52	94.11	91.17

The results of testing accuracy are much different from those of LOO accuracy. First, accuracy varies a lot with the number of features while the change is little in LOO accuracy. Second, SVM-RFE and AROM-SVM are not much superior to the other two methods when considering the top10 features. Third, all algorithms have much higher LOO accuracy than the testing accuracy. In terms of accuracy using top10 features, t-test and ‘fisher’ are better than SVM-RFE and AROM-SVM in this experiment. If we compare them using the accuracy of top20 features, SVM-RFE is the best (97.05%) and AROM-SVM’s accuracy is the second (88.23%) and t-test and ‘fisher’ are the worst (73.52%).

Since using how many features for classification is an open problem, people in general use top10, top20 or top50 features in data analysis. Now it is hard to tell whether one method is better than another because the performance of a method

varies with the validation measures. Using different validation measures, we sometimes get conflict conclusions. The observations obtained from LOO accuracy are doubtful. The experimental results indicate that testing accuracy is more helpful to compare the performance of the four feature ranking algorithms than LOO accuracy.

From the experimental results in Table 1 and Table 2, some rough trends can be observed. However, how many features should be used to achieve the best accuracy is still unknown. To tackle this problem, FSRR uses the feature similarity to remove redundancy, thus select a suitable feature set for better classification. The testing accuracy of FSRR using three different similarity measures is demonstrated in Table 3.

Table 3. Testing accuracy of FSRR using three different similarity measures on Prostate cancer data.

	t-test	fisher	SVM-RFE	AROM-SVM
CC	92.65	92.65	69.12	92.65
LSRE	94.12	94.12	72.79	91.92
MICI	91.91	91.91	79.42	92.65

All the accuracy is obtained by using three features (genes).

CC, LSRE, and MICI are the three feature similarity measures introduced in section 3.3. All the accuracy in Table 3 is obtained by using three features. The highest accuracy of each method is in bold. From Table 3, we can see that the three similarity measures are all helpful to t-test, 'fisher', and AROM-SVM and result in much higher accuracy in the three algorithms than in SVM-RFE. Comparing the results in both Table 2 and Table 3, we can see that all of the three similarity measures improve the accuracy of t-test and 'fisher'. Especially, LSRE is the best of the three similarity measures. When applying CC to SVM-RFE, it gives the lowest accuracy(69.12%). The experiment results demonstrate that FSRR is robust when working with t-test, 'fisher', and AROM-SVM while cannot improve the performance of SVM-RFE. Since FSRR is an unsupervised feature selection algorithm, the most appropriate way to use FSRR is to let it work with t-test or 'fisher' such filter-model feature ranking algorithms.

4. CONCLUSION

In this paper, we propose an algorithm which reduces the redundancy by feature similarity measures. Three feature similarity measures are presented and tested on four feature ranking algorithms (t-test, fisher, SVM-RFE, and AROM-SVM) with gene expression data. Our experimental results demonstrate that reducing redundancy by feature similarity can select very small feature set and achieve good performance in working with t-test and 'fisher'. It is efficient and effective for unsupervised feature selection.

REFERENCES

- [1] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*,3:1157–1182,2003.
- [2]Liu, H., J. Li, L. Wong, A comparative study on feature selection and classification methods using gene expression profiles and proteomic pattern, *Genomic Informatics*, 13, 51-60, 2002.
- [3] C. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press,1996.
- [4] I. Guyon, Gene selection for cancer classification using support vector machines, *Mach. Learn.*, vol. 46, pp. 389–422, 2002.
- [5] Weston, J., Elisseeff, A., Scholkopf, B. and Tipping, M., Use of the Zero-Norm with Linear Models and Kernel Methods, *Journal of Machine Learning Research*, 3, 1439-1461. 2003.
- [6] P. Mitra, C.A. Murthy, and S.K. Pal, Unsupervised feature selection using feature similarity, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 3, 301–312, 2002.
- [7] C.L. Blake and C.J. Merz, *UCI repository of machine learning databases*, 1998.
- [8] S.K.Das, Feature Selection with a Linear Dependency Measure, *IEEE Trans. Computers*. 1051-1054, 1971.
- [9]V. Vapnik. Estimation of dependencies based on empirical data. Springer series in statistics. Springer, 1982.
- [10]W. S. Noble, Support vector machine applications in computational biology, *Kernel Methods in Computational Biology*. B. Schoelkopf, KTsuda and J.-P. Vert, ed. MIT Press, 71-92, 2004.
- [11]B. Schölkopf, I. Guyon, and J. Weston, *Statistical Learning and Kernel Methods in Bioinformatics, Artificial Intelligence and Heuristic Methods in Bioinformatics 183*, (Eds.) P. Frasconi und R. Shamir, IOS Press, Amsterdam, The Netherlands,1-21, 2003
- [12] J.Weston, A. Elisseeff, G. Bakir, and Fabian Sinz, *The spider for matlab - v1.4*, 2004.