A Fast Supervised Method of Feature Ranking and Selection for Pattern Classification

Suranjana Samanta and Sukhendu Das

Dept. of CSE, IIT Madras, Chennai, India
ssamanta@cse.iitm.ac.in, sdas@iitm.ac.in

Abstract. This paper describes a fast, non-parametric algorithm for feature ranking and selection for better classification accuracy. In real world cases, some of the features are noisy or redundant, which leads to the question - which features must be selected to obtain the best classification accuracy? We propose a supervised feature selection method, where features forming distinct class-wise distributions are given preference. Number of features selected for final classification is adaptive, but depends on the dataset used for training. We validate our proposed method by comparing with an existing method using real world datasets.

1 Introduction

Feature selection is an important pre-processing step for classification of any large dimensional dataset. The noisy features confuse the classifier and often mislead the classification task. In order to obtain better accuracy, the redundant, inappropriate and noisy features should be removed before the process of classification. Given a dataset of D dimension, the aim of feature selection is to select a subset of best M features ($M < D$), which produces better result than that of the entire set. A lot of work has been done on feature ranking and selection [1], [2]. Feature selection methods can be broadly classified into two types [2], filter and wrapper methods. In filter method, features are first ranked based on a specific criteria (mostly statistical evaluation), and then the best 'M' features are considered for classification. Huan Liu and Lei Yu [2] have explained the feature selection algorithm into four different steps, namely, subset generation, subset evaluation, stopping criteria and result validation. In most of the work done earlier adaptiveness in the final number of selected features is missing, where the input is either a pre-determined number of features to be selected or a pre-determined threshold [1], which when exceeded stops the selection algorithm.

We propose a new feature ranking method and an adaptive feature selection technique. First, we obtain a score (quality measure) using any of the three proposed measures for each feature (dimension) based on the class-wise scatter of the training samples. Next, we rank the features using a greedy search method. Finally, we determine the number of top ranked features to be used for final classification, using Fisher's discriminant criteria. Results obtained over real world datasets show significant improvement in both performance and speed

S. Chaudhury et al. (Eds.): PReMI 2009, LNCS 5909, pp. 80–85, 2009.
© Springer-Verlag Berlin Heidelberg 2009

when compared with one of the current state of the art [1]. The next few sections give a detail discussion of the different steps of the algorithm.

2 Feature Selection and Ranking

The overall process consists of two steps - feature ranking and selecting the appropriate number of features in the final set. For speedup in computation a greedy method has been used, where we assume the top M ranked features to give better classification result than the total set.

2.1 Feature Ranking

A good feature subset should have low within-class scatter and large between-class scatter. In an ideal case, instances of a good feature or a subset of features belonging to a particular class, should form a compact non-overlapping cluster. In such a case, we expect C non-overlapping clusters Cr_j, $\forall j = 1, 2, ..., C$, where C is the number of classes. We obtain a class label, $\psi(Cr_j)$, for each of the C clusters formed, where $j = 1, 2, ..., C$. Let $class(i)$ denotes the class label of the i^{th} training sample. To calculate the $\psi(Cr_j)$ for a cluster Cr_j $(j = 1, 2, ..., C)$, *we find the class label from training data, which has the highest number of instances in* Cr_j. In this way all clusters are approximated by the distribution made by instances of training samples belonging to each of the distinct classes.

The method for computing $\psi(Cr_j)$, $\forall j = 1, 2, ..., C$, depends of the clustering algorithm used, which has been implemented in two ways. One way of clustering is to fit a 1-D Gaussian function separately on the data instances belonging to each of the classes using GMM (Gaussian Mixture Model). The class assignment for a cluster is the class label for instances of the training data used to fit the Gaussian curve. The other way of clustering is FCM (Fuzzy C-Means), an unsupervised algorithm, for C clusters. To compute the $\psi(.)$ for C clusters in this case, we form a confusion matrix (CM), where we calculate the number of instances of different classes (as given in training data) present in each cluster. The rows of CM correspond to the C clusters formed using FCM and the columns of CM corresponds to the C classes of the training dataset. Computation of ψ involves: *(i)* Convert the elements of $CM(C \times C)$ to a probability, using number of samples in each class; *(ii)* Obtain a class label ψ for each cluster, having maximum probability (row-wise) in a cluster.

After obtaining the ψ for each cluster, we determine the probability (soft class labels) of each instance belonging to each of the C clusters given by $p(k|j)$, $k = 1, 2, ..., N$ and $j = 1, 2, ..., C$; to incorporate fuzziness in the ranking process. Based on this clustering result, we calculate $P(k) = max(p(k|j))$, $\forall k = 1, 2, ..., N$, where N is the total number of training samples. We find a score for each feature (dimension) based on its class-wise distribution in each cluster. We assign a penalty for the occurrence of all instances i belonging to Cr_j whenever $class(i) \neq \psi(Cr_j)$. Class information is used to obtain how many instances of all other classes exist in a cluster. We calculate a probabilistic score based on the concept of Cohen-Kappa (κ) measure [3], which is used to measure inter-classifier agreement. Though we are not dealing with multiple classifiers here, we calculate the agreement between original class label of the training data with the class labels (ψ) assigned by clustering, using the Cohen-Kappa measure. This measure gives an indicator of how well the clusters are formed according to the original class-labels. So, we calculate the first measure: S_1, as,

$$S_1 = (c_1 - c_2)/(N - c_2) \tag{1}$$

where, $c_1 = |\{k|class(k) = \psi(Cr_j),\, k \in Cr_j \,\forall j = 1, 2, ..., C\}|$ and $c_2 = |\{k|class(k) \neq \psi(Cr_j),\, k \in Cr_j \,\forall j = 1, 2, ..., C\}|$. We extend this idea to calculate another fuzzy measure: S_2, as,

$$S_2 = (\phi_r - \phi_c)/(N - \phi_c) \tag{2}$$

where, ϕ_r and ϕ_c are similar to relative-agreement and chance-by-agreement of Cohen-Kappa [3] which is given by, $\phi_r = \sum_{j=1}^{C} \sum_{k \in Cr_j} P(k)|_{class(k)=\psi(Cr_j)}$ and $\phi_c = \sum_{j=1}^{C} \sum_{k \in Cr_j} P(k)|_{class(k)\neq\psi(Cr_j)}$.

The third measure: S_3 is based on the principle that the matrix CM should be diagonal in an ideal case which can be expressed as,

$$S_3 = (1 + [sum(CM) - trace(CM)])^{-1} \tag{3}$$

The last stage of the feature ranking process involves sorting the feature dimension sequentially based on any one of the measures (S_1, S_2 or S_3). At any stage of iteration, let F_s be the set of selected features and F_{us} be the remaining set of features. Once a feature is ranked, it is removed from F_{us} and is added to Fs. The feature with the highest score is ranked first. The next ranked feature should not only have good score but also should be distinct or non-correlated with the already selected features in order to contribute to better classification accuracy. Hence, for obtaining the next feature to be ranked, we consider the correlation between all pairs of features f_t and f_s, where $f_t \in F_{us}$ and $f_s \in F_s$. For obtaining the next ranked features from F_{us}, we compute the difference in the score calculated of the feature and the mean of the correlation of that feature with other set of features selected in F_s. The feature with the maximum value is chosen as the next ranked feature. This process goes on until $|F_s| = D$ and $|F_{us}| = NULL$. The total method has been described in Algorithm 1.

2.2 Feature Selection

After the process of feature ranking, the best M features must be selected. We choose the popular Fisher's discriminant criteria [4], for selecting a subset of M features from D rank-ordered features ($M < D$), which is given by $trace(S_w^{-1} \times S_b)$ where, S_b and S_w are the between-class and within-class scatter matrix [4]. Fisher's criteria is computed for the top ranked features, at each iteration with increasing dimension. This criteria when observed with increasing number of features in the dataset, shows an abrupt change in the gradient at a point. The

Algorithm 1. Algorithm for ranking the feature set.
INPUT: Training dataset $'data'$ along with their class-labels.
OUTPUT: Rank-ordered dataset F_s.

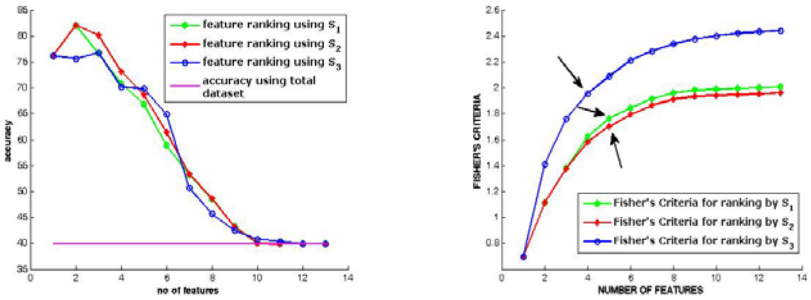
```
for $i = 1$ to D do
    Form $C$ clusters $(Cr)$ with the $i^{th}$ feature dimension in $'data'$, using FCM or GMM.
    Form confusion matrix $(CM^{C \times C})$ where, $cm_{jk} \in CM$ denotes the number of samples belonging to $j^{th}$ class and present in $k^{th}$ cluster $(Cr_k)$.
    Compute $\psi$ (class label) for each cluster, using $CM$ in case of FCM or class labels in training data in case of GMM.
    Compute $S_i$ using any one of the measures given in Eqns. 1, 2 or 3.
end for
$F_s \leftarrow data[:, \arg\max_i(S_i)]$ /* select the best dimension */
$F_{us} \leftarrow [data - F_s]$
while $|F_{us}| \neq NULL$ do
    for $i = 1$ to $|F_{us}|$ do
        $h = \sum_{i=1}^{|F_{us}|} \sum_{j=1}^{|F_s|} corr(f_i, f_j)$ where, $f_i \in F_{us}$, $f_j \in F_s$
        $r_i = S_i - h/|F_s|$
    end for
    $k = \arg\max_i [r_i]$
    Remove $k^{th}$ feature from $F_{us}$ and insert as last dimension in $F_s$
end while
```

features which are ranked after this point of transition are generally redundant and confusing, and hence removed from the final dataset.

Fig. 1(a) shows the change in the classification accuracy of Wine dataset obtained from UCI repository [5]. Fig. 1(b) shows the plot of Fisher's criteria with increasing number of rank-ordered features in the subset. The arrow marks points to the 'knee point' of the curves, for each of the scores, where there is a significant change in the gradient of the curve. In Fig. 1(b), after considering the 4^{th} rank-ordered feature the gradient of the curve does not show any appreciable change, and hence we select the first 4 features, which is consistent with the drop in accuracy in Fig. 1(a).

3 Experimental Results

Experiments done on real world datasets obtained from UCI Repository [5], show significant improvement of classification accuracy after feature ranking and selection. One-third of the training samples are used for feature ranking and selection and the rest are used for testing. Results of accuracy in classification obtained using SVM with Gaussian kernel have been reported after ten fold cross validation. Fig. 2(a) and (b) show four results each, obtained using FCM and GMM clustering method respectively using S_1 (green curve), S_2 (red curve),



(a) (b)

Fig. 1. Plot of (a) classification accuracy & (b) Fisher's criteria with increasing number of rank-ordered features. Wine Dataset has been used from UCI Repository [5].

Table 1. Accuracy and computational time of the proposed feature ranking and selection method using both FCM & GMM clustering. (best result in bold)

Dataset	Total %	Accuracy %						Time (in secs.)		
		FCM			GMM			FCM	GMM	M.I. [1]
		S_1	S_2	S_3	S_1	S_2	S_3			
Cancer	86.77	92.70	**93.08**	92.90	92.00	91.54	92.53	**0.20**	1.37	14.84
Lymph	57.45	67.98	**78.51**	72.55	70.21	66.06	67.55	**0.39**	0.76	3.23
Metadata	29.09	69.83	**100**	**100**	**100**	**100**	**100**	14.20	**3.35**	542.76
Spectf	52.88	63.58	63.65	64.81	64.81	**66.54**	65.38	**3.60**	4.19	90.64
Vehicle	26.19	29.66	29.66	31.17	33.63	35.20	**37.14**	**1.44**	6.81	1669
Zoo	80.31	89.84	94.84	**96.72**	95.94	95.47	92.81	**0.55**	0.61	2.07

S_3 (blue curve) and mutual information (mustard curve) [1] as feature ranking measures. The magenta horizontal line shows the classification accuracy using total dataset (baseline). Table 1 gives a comparative study of the classification accuracy obtained using two different clustering methods along with the baseline accuracy using total dataset. It also shows the average time taken (in seconds) to compute the feature ranking and selection using any of the proposed measures (S_1, S_2 or S_3), with respect to the method using Mutual Information [1].

4 Conclusion

We have observed the performance of the feature selection algorithm on real world datasets. Most of the existing feature selection algorithms are time consuming and parametric. To overcome this problem, we propose a fast nonparametric selection algorithm which improves the classification accuracy. Results discussed in the paper show that the proposed feature ranking method is better than in [1].

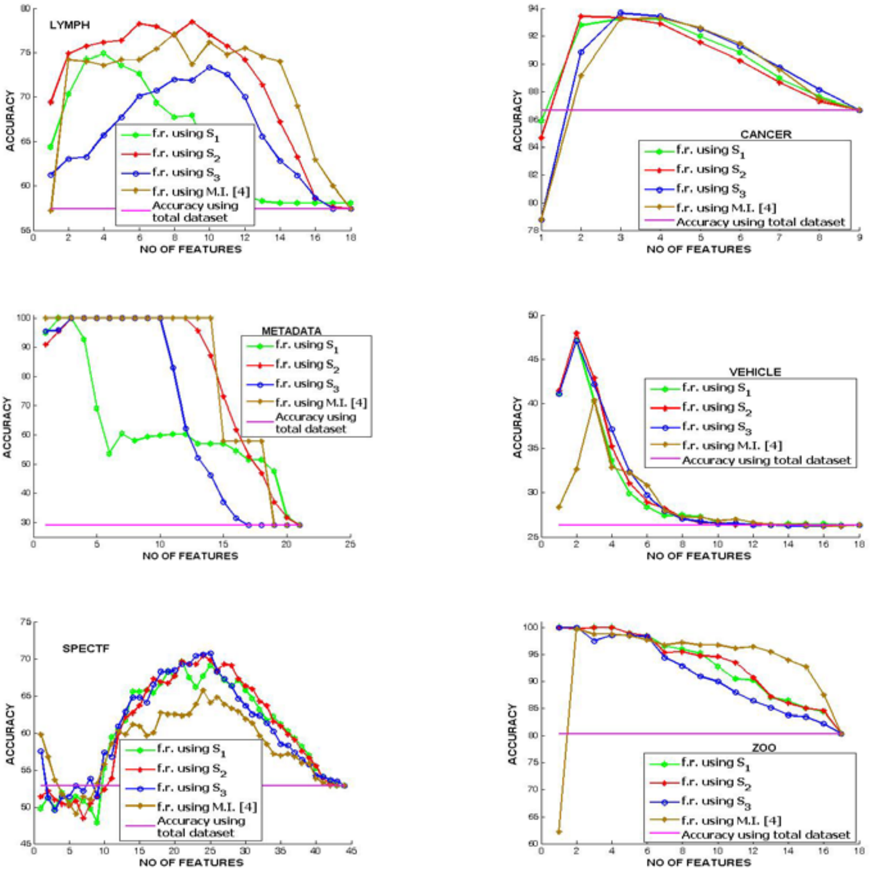


Fig. 2. Comparison of performance of the proposed method with [1], using (a) FCM- and (b) GMM based clustering. * f.r.: feature ranking; M.I.: mutual information [1].

References

1. Guoa, B., Damper, R., Gunna, S.R., Nelsona, J.: A fast separability-based feature-selection method for high-dimensional remotely sensed image classification. Pattern Recognition 41, 1653–1662 (2007)
2. Liu, H., Yu, L.: Toward integrating feature selection algorithms for classification and clustering. IEEE Transactions on Knowledge and Data Engineering 17, 491–502 (2005)
3. Sim, J., Wright, C.C.: The Kappa statistic in reliability studies: Use, interpretation, and sample size requirements. Physical Therapy 85, 257–268 (2005)
4. Fukunaga, K.: Introduction to Statistical Pattern Recognition, 2nd edn. Computer Science and Scientific Computing Series. Academic Press, London (1990)
5. Asuncion, A., Newman, D.: UCI machine learning repository (2007)