Pattern Recognition 41 (2008) 3706 – 3719

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Feature selection using localized generalization error for supervised classification problems using RBFNN

Wing W.Y. Ng[a,b,*], Daniel S. Yeung[a,b], Michael Firth[c], Eric C.C. Tsang[b], Xi-Zhao Wang[d]

[a] *School of Computer Science and Engineering, South China University of Technology, Guangzhou 510640, China*
[b] *Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong*
[c] *Department of Finance and Insurance, Lingnan University, Hong Kong*
[d] *Machine Learning Center, Faculty of Mathematics and Computer Science, Hebei University, Baoding 071002, China*

ARTICLE INFO

Article history:
Received 1 March 2007
Received in revised form 1 March 2008
Accepted 5 May 2008

Keywords:
Feature selection
Neural network
Generalization error
RBFNN

ABSTRACT

A pattern classification problem usually involves using high-dimensional features that make the classifier very complex and difficult to train. With no feature reduction, both training accuracy and generalization capability will suffer. This paper proposes a novel hybrid filter–wrapper-type feature subset selection methodology using a localized generalization error model. The localized generalization error model for a radial basis function neural network bounds from above the generalization error for unseen samples located within a neighborhood of the training samples. Iteratively, the feature making the smallest contribution to the generalization error bound is removed. Moreover, the novel feature selection method is independent of the sample size and is computationally fast. The experimental results show that the proposed method consistently removes large percentages of features with statistically insignificant loss of testing accuracy for unseen samples. In the experiments for two of the datasets, the classifiers built using feature subsets with 90% of features removed by our proposed approach yield average testing accuracies higher than those trained using the full set of features. Finally, we corroborate the efficacy of the model by using it to predict corporate bankruptcies in the US.

1. Introduction

With the availability of fast computers, broadband Internet, and cheap, high capacity storage, datasets have become ever larger. Usually, domain knowledge and personal bias influence the choice of features. Although these parameters may not fully describe the problem, some parameters may be included just for fear of losing something useful. When the number of parameters (input features) of the dataset becomes large, the pattern classification systems trained for differentiating the sample points into different classes also get more complex. On the other hand, if it is not necessary to collect so many input features, the cost of data collection and storage will be reduced.

A major problem in pattern classification is how to build a simple classifier that has good performance. By "good performance" we mean a system that can be quickly trained, is highly accurate and responds quickly to future unseen samples, and is easily understood by people. Perhaps the most straightforward way to reduce the complexity of a classifier is to reduce the number of input features.

Given the training dataset $D = \{(\mathbf{x}_b, F(\mathbf{x}_b))\}_{b=1}^{N}$ consisting of N training samples $(\mathbf{x}_b)$ with F denoting the unknown input–output mapping of the classification problem that one would like to approximate using a classifier (e.g. a neural network), the training error (R_{emp}) and generalization error (R_{true}) for the entire input space (T) of the classifier f_θ are defined as

$$R_{emp} = \frac{1}{N}\sum_{b=1}^{N}(F(\mathbf{x}_b) - f_\theta(\mathbf{x}_b))^2 \tag{1}$$

$$R_{true} = \int_T (F(\mathbf{x}) - f_\theta(\mathbf{x}))^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x} \tag{2}$$

where $p(\mathbf{x})$ denotes the true unknown probability density function of $\mathbf{x}$, and θ denotes the set of parameters in the classifier f_θ. The ultimate goal of training a classifier is to minimize the generalization error for unseen samples (i.e. minimizing the differences between the real unknown input–output mapping function and the mapping approximated by f_θ). Moreover the ultimate goal of feature selection is to maintain the classifier's generalization capability using a reduced

[*] Corresponding author at: School of Computer Science and Engineering, South China University of Technology, Guangzhou 510640, China.
E-mail addresses: wingng@ieee.org (W.W.Y. Ng), csdaniel@comp.polyu.edu.hk (D.S. Yeung), mafirth@ln.edu.hk (M. Firth), csetsang@comp.polyu.edu.hk (E.C.C. Tsang), wangxz@mail.hbu.edu.cn (X.-Z. Wang).

0031-3203/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2008.05.004

set of features. Classifiers (e.g. neural networks) are usually not expected to recognize unseen samples that are too different from the training samples. Therefore, assessing the generalization capability of a classifier for those unseen samples may be counter-productive to classifier learning. So, Ng et al. [1,2] proposed a localized generalization error model for bounding the generalization error (R_{SM}^*) for a classifier for unseen samples similar to the training samples. In our proposed feature selection method (R_{SM}FS), we remove the feature subset that yields the smallest contribution to the R_{SM}^*. In terms of probability, the classifier trained using the reduced feature subset will not lose its generalization capability if R_{SM}^* remains unchanged. In this paper, the widely adopted radial basis function neural networks (RBFNNs) with Gaussian basis function [3,4] will be used to demonstrate the R_{SM}FS method.

A brief literature review is presented in Section 2. In Section 3 we describe the localized generalization error model. The novel feature selection method R_{SM}FS is presented in Section 4, while experimental results are shown in Section 5. Section 6 concludes the paper.

2. Existing feature selection methods

Broadly speaking, the number of input features is reduced using three feature selection approaches: filters, wrappers, and embedded [5–7]. Under certain circumstances in the learning process of a decision tree, some features are ignored in the final decision tree if they have a minor influence on the classification [8]. This is a special case of feature selection and we will not discuss it in this work. In the following two sub-sections, we will introduce the filter and wrapper approaches.

In Fig. 1, we illustrate the relationship between different relevant measures for feature selection. A relevant measure is employed in each feature selection method and we will have more discussion on each of these measures in Sections 2.1 and 2.2.

Principal component analysis [9,10] and other transformation-based feature reduction methods are not discussed in this paper because they do not select the features from the original feature set. These methods transform the feature set into a lower-dimensional feature vector by combining several features. Transformation-based feature reduction methods do not reduce the cost of future sample collection and storage. Moreover, the newly created feature vector is usually difficult to interpret. For example, in the physiology field, a feature vector may be composed of blood pressure times the square of body height and this kind of feature does not help people understand the problem.

2.1. Filter approaches

Filter approaches make use of statistical information of the dataset to carry out feature selection and are independent of the classification system. These approaches rely on the definition of a relevant measure.

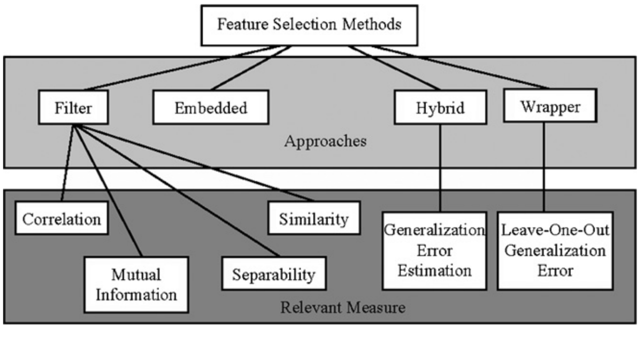

Fig. 1. Relationship between relevant measures for feature selection.

The simplest measure may be the correlation between the input and output using the correlation coefficient [6,7]. The absolute value of the correlation coefficient may be used because we may want to focus on the magnitude of the correlation between the input feature and the output. The major drawback is that it ignores any nonlinear correlation between input and output.

This problem could be solved by using the mutual information to replace the correlation coefficient. In mutual information approaches [11,12], the mutual information between the inputs and outputs are computed and sorted. The input feature that yields the maximum mutual information to the outputs is selected. Then the mutual information between the outputs and also between the selected feature subsets is computed. The feature yielding the maximum mutual information is added into the selected subset. These procedures are repeated until a specified number of features are reached. This approach has a sound theoretical underpinning, yet the computation of the probability density function between features and outputs is expensive. Kwak et al. [13] improved the approach by using the Parzen Window to estimate the density functions, but the computation effort is still high for a dataset with large numbers of features and samples In this work, we adopt the definition of mutual information proposed in Ref. [11].

Mitra [14] proposed using a similarity measure between input features. In his work, features are grouped by similarity and only one feature in each group is selected. This method does not take into account the performance of the features and simply deletes similar features. As a result, the performance of the feature selection is determined by choosing the number of groups and the similarity measure.

The authors in Refs. [15,16] applied the class separability measure to select feature subsets. The feature with the maximum class separability is selected first. The next feature with the maximum class separability will then be selected after removing the first one. The process stops when no more features provide class separability larger than a given threshold. This approach considers the training classification accuracy indirectly. However it cannot deal with the case in which the samples from one class are surrounded by samples from another class. In Ref. [17], the authors proposed removing features that are inconsistent to the class label (i.e. could not separate samples from two classes). However, the point-wise comparison makes the method infeasible for a large dataset.

One may observe that the above filtering approaches require users to determine the number of features selected, rather than providing stopping criteria. Furthermore, the generalization performance of the classifier is not a consideration of the filtering feature selection methodologies, even though it is the ultimate goal of building classifiers. Yet, they are free from the bias of classifier training. The feature selection criteria presented in this section relate to the training accuracy indirectly.

2.2. Wrapper approaches

Wrapper feature selection methodologies combine both the feature selection and output of the classification system into a single system [18]. Most of the wrappers employ the Leave-One-Out searching method [19] which, in each step, evaluates the training accuracy when one of the features is left out, and then removes the feature yielding the least reduction in accuracy. The Leave-One-Out wrapper feature selection methodology can be applied to any classification system.

However, the procedures mentioned above only make use of the training accuracy as the relevant measure. Since a classifier is built, the validation accuracy is used to ensure that the classification