Reduced Kernel Regression for Fast Classification

Fabian Hoti
Rolf Nevanlinna Institute
P.O. Box 4 (yliopistonkatu 5)
FIN-00014 University of Helsinki
Finland

Tel: +358-9-191 22770
Fax: +358-9-191 22779
E-mail: hoti@helsinki.fi

Keywords: Pattern recognition; Kernel regression; Radial basis function

26 January 1998

1 Introduction

Pattern recognition has applications in various fields such as speech recognition and classification of handwritten characters. Discrimination techniques used in pattern recognition can be divided into parametric and non-parametric according to their structures. In this paper we introduce a method that lies somewhere between these two categories. First we will introduce the basics of classification and the theoretically optimal Bayes classifier. Next we will use a kernel regression method to approximate the bayes classifier and finally we will reduce its complexity considerably. The resulting method can then be viewed either as a local parametric method or as a non-parametric radial basis function expansion familiar from neural network studies.

The performance of the method is tested with two real world data and the results are compared to those gained from popular classifiers [1]. It appears that by using our method one can get competitive classification results ten times faster. Here we measure the speed by the complexity of the methods, which in these cases depends directly on the number of kernels used.

2 Statistical Classification

Let $x = [x_1, \cdots, x_d]^T \in \mathcal{R}^d$, be a pattern vector, a multidimensional measurement, taken from an object that belongs to one of c different classes. Given a pattern vector x, a classification problem is to guess from which class $j \in \{1, \cdots, c\}$ the measurement originated. A classifier can now be regarded as a function $g : \mathcal{R}^d \to \{1, \cdots, c\}$.

In the sense of probability theory the whole observation is a $d+1$ dimensional random vector $[X^T, J]^T$. Let $P_j = P(J = j)$ be the *a priori* probability and f_j the probability density of class j. Now the density of the pooled data is $f = \sum_{j=1}^{c} P_j f_j$ and the *a posteriori* probability of class j conditional on $X = x$ is $q_j(x) = P_j f_j(x)/f(x)$, if $f(x) \neq 0$. The optimal classifier in the sense that it

minimizes the probability of misclassification is called the Bayes classifier. It can be shown to be given by

$$q_{BAYES}(x) = \underset{j=1,\cdots,c}{\operatorname{argmax}} q_j(x).$$

In case j is not unique, the chose can be made freely among the classes with the biggest value q_j. In practice classifiers are constructed either by estimating the class densities or by using some regression technique to directly estimate the *a posteriori* probabilities from given data.

3 Reduced Kernel Regression

Let $\{(X_1, J_1), \cdots, (X_n, J_n)\}$ be a sample. We estimate the class density with the *kernel estimator*

$$\hat{f}_j(x) = n_j^{-1} \sum_{i=1}^{n} Y_j^i K_h^i(x - X_i),$$

where $n_j = \#\{i : J_i = j\}$, Y_j^i is 1 if $J_i = j$ and Y_j^i is 0 otherwise. And $K_h(x) = h^{-d}K(x/h)$, where K is a function, which integrates to one and is symmetric about the origin, and h is a smoothing parameter. Further we derive an estimate for the j'th class *a posteriori* probability as follows

$$\begin{aligned} q_j(x) = \frac{P_j f_j(x)}{f(x)} &\approx \frac{\hat{P}_j \hat{f}_j(x)}{\sum_{k=1}^{c} \hat{P}_k \hat{f}_k(x)} \\ &= \frac{\frac{n_j}{n}\frac{1}{n_j}\sum_{\{i:J_i=j\}} K_h(x - X^i)}{\sum_{k=1}^{c} \frac{n_k}{n}\frac{1}{n_k}\sum_{\{r:J_r=k\}} K_h(x - X^r)} \\ &= \frac{\sum_{i=1}^{n} Y_j^i K_h(x - X^i)}{\sum_{r=1}^{n} K_h(x - X^r)} = \hat{q}_j(x). \end{aligned}$$

This estimator is known as the Nadaraya – Watson estimator [4]. Note that the estimate is a weighted sum over Y_j^i and can be written as

$$\hat{q}_j(x) = \sum_{i=1}^{n} Y_j^i w_i(x), \tag{1}$$

where $w_i(x) = K_h(x - X^i) / \sum_{r=1}^{n} K_h(x - X^r)$ and $\sum_{i=1}^{n} w_i = 1$. We can further improve the flexibility of the estimator by replacing the constants Y_j^i with locally fitted functions, for example polynomials of different degree. In our studies we only considered constants and first-order polynomials

$$\hat{q}_j(x) = \sum_{i=1}^{n} (a_j^i - (x - X_i)^T b_j^i) w_i(x). \tag{2}$$

The fitting was done in the least square sense

$$\sum_{j=1}^{c} \sum_{i=1}^{n} (\hat{q}_j(X_i; a, b) - Y_j^i)^2 = \min_{a,b}!,$$

where $a, b = (a_j^i, b_j^i), i \in \{1 \cdots n\}, j \in \{1 \cdots c\}$. The resulting scheme is a very effective estimator, but as the dimension of the data increases it is due to run into trouble. The complexity of the estimator depends directly on the number of data

points used and in higher dimensions the amount of data needed to get good estimates increases rapidly. This phenomenon is known as the curse of dimensionality.

We attacked the problem of dimensionality by adapting some methods from neural network studies. The Radial basis function expansion of Moody and Darken [2] has the form

$$\hat{q}_j(x) = \sum_{r=1}^{M} \frac{a_j^r K(\frac{x-m_r}{h})}{\sum_{s=1}^{M} K(\frac{x-m_s}{h})}. \tag{3}$$

Here $M << n$ kernels with centers at locations m_r are used to produce a weighted sum of the constants a_j^r. This scheme is fast to use and its complexity does not depend on the amount of data. Now by combining the flexibility of local fitting in scheme (2) and the speed achieved in scheme (3) we get the Reduced Kernel Regression scheme

$$\hat{q}_j = \sum_{r=1}^{M} (a_j^r - x^T b_j^r) w_i(x), \tag{4}$$

where $w_r(x) = K(\frac{x-m_r}{h_r}) / \sum_{s=1}^{M} K(\frac{x-m_s}{h_s})$. Here the constants a_j^r, b_j^r, kernel centers m_r and smoothing parameters h_r are chosen so that the least square error is minimized. A similar scheme (4) was proposed in [3].

4 Case Studies and Conclusions

We tested the performance with two different data sets: a highly non-normal, 2-class, 5-dimensional phoneme data set and a 64-dimensional handwritten digit data set with 10 roughly normally distributed classes. We compared the results with those obtained in [1]. The results show that in both cases by using our method one can reduce the complexity considerably, compared to traditional non-parametric methods, without loosing classification accuracy.

References

[1] L. Holmström, P. Koistinen, J. Laaksonen, and E. Oja. Neural and statistical classifiers—taxonomy and two case studies. *IEEE Transactions on Neural Networks*, 8:5–17, 1997.

[2] J. Moody and C. Darken. Fast learning in networks of locally–tuned processing units. *Neural Computation*, 1:281–294, 1989.

[3] K. Stokbro, D.K. Umberger, and J.A. Hertz. Exploiting neurons with localized receptive fields to learn chaos. *Complex Systems*, 4:603–622, 1990.

[4] M.P. Wand and M.C. Jones. *Kernel Smoothing*. Chapman & Hall, 1995.