Report on Learning Methods

Deliverable D 5.3

Month 12

Project Start:	01/02/2007
Project Duration:	36 Months
Priority area	2.3.6
Contract No.:	FP6-045547
Website:	

Due-Date:	16/06/2009
Delivery:	16/06/2009
Lead Partner:	University of Surrey
Project Leader	UvA
Dissemination Level:	Public
Status:	Draft
Approved:	
Version:	1

Deliverable 5.3: Learning tools, second version

Muhammad Atif Tahir, Josef Kittler, Krystian Mikolajczyk, Fei Yan

June 8, 2009

1 Introduction

The main objective of this report is the description of how to use Kernel Discriminant Analysis using Spectral Regression (SR-KDA) [2, 3]. Details of this algorithm can be found in Deliverable 5.1 and briefly discussed in Section 2. This method is successfully used in various benchmark data sets such as Mediamill Challenge and Pascal VOC and also ranked first in Pascal VOC 2008 competition.

The software consists of two main parts. The first part is the training phase while second part is testing phase. Both parts are implemented in C++. It should be noted that training is traditionally a off-line procedure while testing is a on-line procedure. Please note that this deliverable should be read in conjunction with Deliverable 5.2: Learning tools, first version.

2 Kernel Discriminant Analysis using Spectral Regression (SR-KDA)

Kernel Discriminant Analysis is a nonlinear extension of LDA which maps the original measurements into a higher dimensional space using the "kernel trick". Let $\mathbf{x}_i$ be training vectors $\mathbf{x}_i \in \mathcal{R}^d, i = 1, \cdots, m$. K is an $m \times m$ kernel matrix. If ν denotes a projective function into the kernel feature space, then the objective function for KDA is

$$\max_{\nu} \quad D(\nu) = \frac{\nu^T C_b \nu}{\nu^T C_t \nu} \tag{1}$$

where C_b and C_t denote the between-class and total scatter matrices in the feature space respectively. Equation 1 can be solved by the eigen-problem $C_b = \lambda C_t$. It is proved in [1] that equation 1 is equivalent to

$$\max_{\alpha} \quad D(\alpha) = \frac{\alpha^T K W K \alpha}{\alpha^T K K \alpha} \tag{2}$$

where $\alpha = [\alpha_1, \alpha_2, ...\alpha_m]^T$ is the eigen-vector satisfying $KWK\alpha = \lambda KK\alpha$. $W = (W_l)_{l=1,....n}$ is a $(m \times m)$ block diagonal matrix of labels arranged such that upper block corresponds to positive examples and lower one to negative examples of the class. Each eigenvector α gives a projection function ν in the feature space.

It is shown in [2] that instead of solving the eigen-problem in equation 2, the KDA projections can be obtained by the following two linear equations

$$\begin{aligned} W\phi &= \lambda\phi \\ (K + \delta I)\alpha &= \phi \end{aligned} \tag{3}$$

where ϕ is an eigenvector of W, I is the identity matrix and $\delta > 0$ is a regularisation parameter. Eigen-vectors ϕ are obtained directly from Gram-Schmidt method. Since $(K + \delta I)$ is positive definite, the Cholesky decomposition is used to solve the linear equations in equation 3. Thus, SR-KDA only needs to solve a set of regularised regression problems [2] and there is no eigenvector computation involved. This results in great improvement of computational cost and allows to handle large kernel matrices.

2.1 Complexity Analysis

The computation of SR-KDA involves two steps: (i) response generation which is the cost of the Gram-Schmidt method (ii) regularised regression which involves solving $(c - 1)$ linear equations using the Cholesky decomposition where c is the number of classes. As in [4], we use the term flam, a compound operation consisting of one addition and one multiplication, to measure the operation counts. The cost of the Gram-Schmidt method requires $(mc^2 - \frac{1}{3}c^3)$ flams. The Cholesky decomposition requires $\frac{1}{6}m^3$ flams and the $c - 1$ linear equations can be solved within m^2c flams. Thus, the computational cost of SR-KDA excluding the cost of Kernel Matrix K is $\frac{1}{6}m^3 + m^2c + mc^2 - \frac{1}{3}c^3$ which can be approximated as [2]

$$\frac{1}{6}m^3 + m^2c \tag{4}$$

Comparing to the cost of ordinary KDA $(\frac{9}{2}m^3 + m^2c)$ [2], SR-KDA significantly reduces the dominant part and achieves a 27 times speedup.

2.2 Complexity Analysis for Visual Category Recognition

As discussed in Section 1, the visual category recognition problem can be formulated as a two class pattern recognition problem. The original data set is divided into N data sets where $Y = \{1, 2, ..., N\}$ is the finite set of concepts. The task is to learn one binary classifier $h_a : X \rightarrow \{\neg a, a\}$ for each concept $a \in Y$. Among the advantages of using SR-KDA in binary classification tasks is that its time complexity scales linearly with respect to N. The total computational cost of SR-KDA for all concepts is

$$\frac{1}{6}m^3 + m^2Nc \tag{5}$$

The above analysis clearly shows the effectiveness of SR-KDA for visual category recognition. In the proposed approach, the main task is to perform Cholesky decomposition and then to solve N linear equations which requires only m^2Nc flams.

3 Usage of the Software

As mentioned in Section 1, both training and testing phases are implemented in C++. Two binary executables are generated: SRKDATrain and SRKDATest. Below we describe how to use that software.

3.1 Training Phase

The program "SRKDATrain" is use to train the classifier. Input to this program are two files. One file consists of Kernel Matrix (Training Data of size $X \times X$) where X is the number of training samples and the other file consists of actual labels $(X \times Y)$ where Y is the number of concepts. The program will return a outputfile consists of matrix of size $(X \times Y)$. This outputfile is required in testing phase. The program also needs a regularisation parameter δ described in Equation 3. The following is the syntax of this program:

1. SRKDATrain.exe labelfilename kernelmatrixtrainingfilename outputfile δ
2. labelfilename: This file consists of labels required in the training phase.
3. kernelmatrixtrainigfilename: Kernel Matrix (Training Data only) store in this file.
4. outputfile: The name of the output file. The program will create a new file and this information is required in testing phase.
5. δ: Regularisation Parameter. Default Value 0.01.

3.2 Testing Phase

The program "SRKDATest" is use to find the confidences for the unknown samples. The higher the value return by this program, the more likely it belongs to the concept class. Input to this program are again two files. One file consists of Kernel Matrix (Test Data of size $X_{New} \times X$) where X is the number of training samples, X_{New} is the number of unknown or test samples. The other file is from the training phase of size $(X \times Y)$ where Y is the number of concepts. The program will return a outputfile consists of scores for each sample within each category (Matrix of size $(X_{New} \times Y)$. The following is the syntax of this program:

1. SRKDATest.exe inputfile kernelmatrixtestingfilename outputfile δ
2. inputfile: The output file from the training phase.
3. kernelmatrixfilename: Kernel Matrix (Test Data only) is stored in this file.
4. outputfile: This file consists of scores for each test sample within each category.

References

[1] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 2(12):2385–2404, 2000.

[2] D. Cai, X. He, and J. Han. Efficient kernel discriminat analysis via spectral regression. In *Proceedings of the International Conference on Data Mining*, 2007.

[3] F. Yan M. A. Tahir, J. Kittler and K. Mikolajczyk. Kernel discriminant analysis using triangular kernel for semantic scene classification. In *In Proceedings of the 7th IEEE International Workshop on Content-Based Multimedia Indexing*, Crete, Greece, 2009.

[4] G. W. Stewart. *Matrix Algorithms Volume I: Basic Decomposition.* SIAM, 1998.