Regression Method for Linear and Kernel Discriminant Learning

V.Vignesh
Assistant professor, Department of Computer Science
Veerammal Engineering College, Dindugul, Tamil nadu, India

G. Naga Nandhini
Department of Information Technology, Sathyabama University, Chennai, Tamil nadu, India

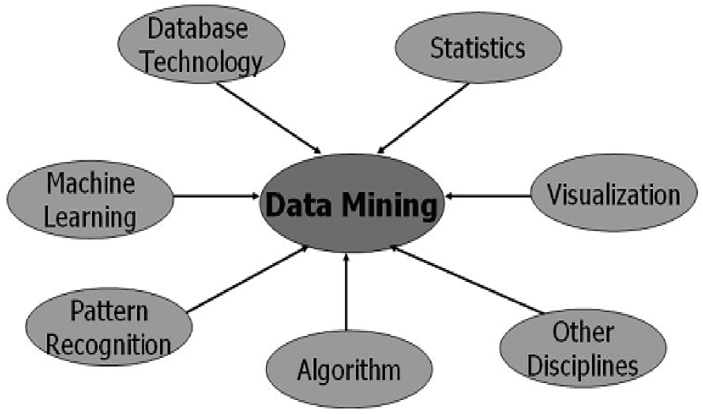
Dr.K.Krishnamoorthy
Professor, Sudharsan Engineering college, Pudukkottai, Tamilnadu, India

Abstract— The main objective of this work is to compare the algorithms of Support Vector Machine(SVM) and Kernel Principal Component Analysis(KPCA) to improve the accuracy and performance of the numerical dataset. SVM fails to consider global information and has high computational complexity. Along with LDA, SVM overcomes the above disadvantages but cannot be applied to non-linear data. Along with KDA, SVM overcomes all the above disadvantages and gives good accuracy. KPCA is used instead of SVM to get better accuracy than SVM.

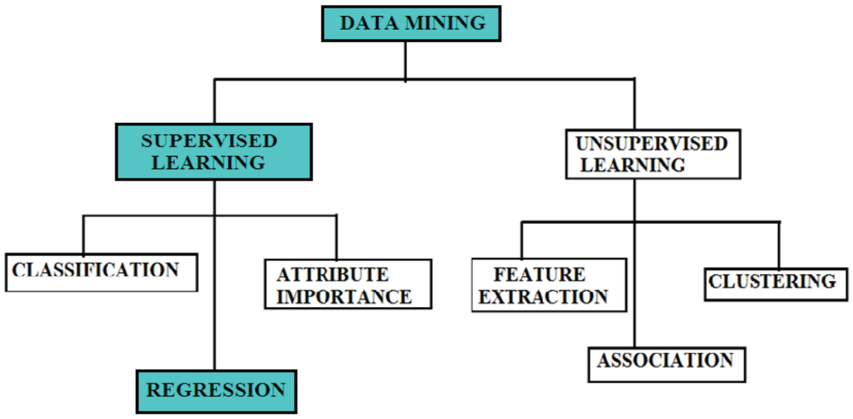
Keywords – SVM, KPCA, LDA, KDA

I. INTRODUCTION

Data Mining, or knowledge discovery, is the computer-assisted process of digging through and analyzing enormous sets of data and then extracting the meaning of the data. They scour databases for hidden patterns, finding predictive information that experts may miss because it lies outside their expectations. Data Mining derives its name from the similarities between searching for valuable information in a large database and mining a mountain for a vein of valuable ore. The contributing areas of Data Mining are information retrieval, high performance computing, machine learning, data visualization. Other areas are neural network, pattern recognition, spatial data analysis, image databases etc.



DATA MINING METHODS



A. REGRESSION

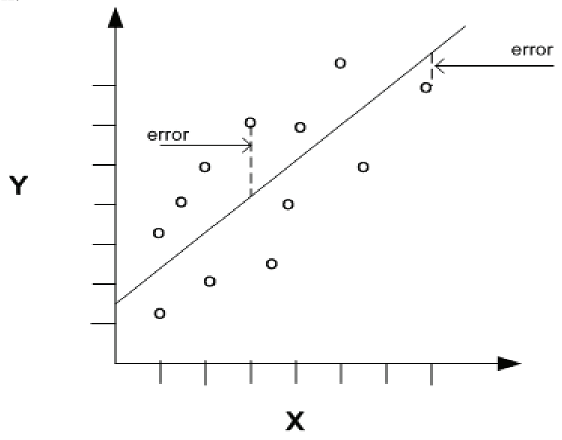
Regression is a supervised learning technique. Regression is the oldest and most well known statistical technique that the Data Mining community utilizes. It deals with continuous quantitative data. In regression, the data can be smoothed by fitting the data to a function.

A regression task begins with a data set in which the target values are known. In the model build (training) process, a regression algorithm estimates the value of the target as a function of the predictors for each case in the build data. These relationships between predictors and target are summarized in a model, which can then be applied to a different data set in which the target values are unknown.

Regression models are tested by computing various statistics that measure the difference between the predicted values and the expected values. The historical data for a regression project is typically divided into two data sets:

- Building the model.
- Testing the model.

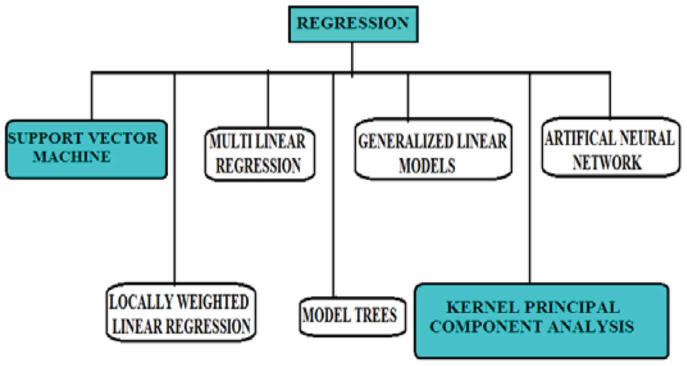
Linear, nonlinear and generalized linear models of regression can be used for prediction. Linear regression is the simplest form of regression.



B. TYPES OF REGRESSION

Different regression algorithms are used to compare the accuracy and performance using different techniques. The Figure shows different types of regression algorithms.

Different types of regression algorithms are:


REGRESSION
SUPPORT VECTOR MACHINE
MULTI LINEAR REGRESSION
GENERALIZED LINEAR MODELS
ARTIFICAL NEURAL NETWORK
LOCALLY WEIGHTED LINEAR REGRESSION
MODEL TREES
KERNEL PRINCIPAL COMPONENT ANALYSIS

C. ALGORITHMS

Support Vector Machine(SVM)

SVM is a set of related supervised learning methods that analyzes data and recognize patterns used for classification and regression analysis. SVM takes a set of input data and predicts for each given input which of two possible classes the input is a member of which makes the SVM a non-probabilistic binary linear classifier. A regression SVM model tries to find a continuous function such that maximum number of data points lie within an epsilon-wide tube around it. SVM is a powerful, state-of-the-art algorithm for linear and nonlinear regression.SVM regression supports two kernels.

SVM also supports active learning. SVM is a machine learning technique that learns patterns based on training data. SVM is a representation of points in space. The goal of SVM approach is to define a hyperplane in a high-dimensional feature space Z, which divides the set of samples in the feature space such that all the points with the same label are on the same side of the hyperplane

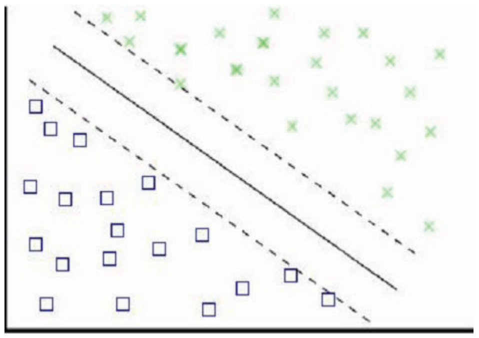


Figure 1: SVM Hyperplane

The model produced by support vector classification only depends on the subset of the training data, because the cost function for building the model does not care about training points lie beyond the margin. The model produced by support vector regression only depends on a subset of the training data, because the cost function for building the model ignores training data that are close to the model prediction.

Kernel Principal Component Analysis

KPCA is an extension of PCA. It is a process of mapping the original data points in to higher dimensional feature space. In KPCA, original operations of PCA are done in a reproducing kernel hilbert space with a non-linear mapping. The KPCA method has been widely used for non-linear feature extraction and data projection. The non-linearity is introduced by mapping the data from the input space to a feature space F.

KPCA utilizes kernel trick to perform operation in the new feature space where data samples are more separable. The kernel trick transforms any algorithm that solely depends on the dot product between two vectors. Wherever a dot product is used, it is replaced with the kernel function. Thus, a linear algorithm can easily be transformed into a non-linear algorithm. This non-linear algorithm is equivalent to the linear algorithm operating in the range space of φ. However, because kernels are used, the φ function is never explicitly computed. This is desirable, because the high-dimensional space may be infinite-dimensional.

According to this approach, the nonlinear data structure in the input space is more likely to be linear after high-dimensional nonlinear mapping. By using a nonlinear kernel function instead of the standard dot product, we implicitly perform PCA in a high-dimensional space F which is non-linearly related to input space. Consequently, KPCA produces features which capture the nonlinear structure in the data better than SVM LDA.

II. SVM WITH LDA

In this module, we implement SVM classification on the dataset regressed from LDA. Classifying data is a common task in machine learning. Suppose some given data points each belong to one of two classes, and the goal is to decide which class a new data point will be in. In the case of SVM, a data point is viewed as a p-dimensional vector (a list of p numbers), and we want to know whether we can separate such points with a $(p-1)$ -dimensional hyperplane. This is called a linear classifier. There are many hyperplanes that might classify the data. One reasonable choice as the best hyperplane is the one that represents the largest separation, or margin, between the two classes. So we choose the hyperplane so that the distance from it to the nearest data point on each side is maximized. If such a hyperplane exists, it is known as the maximum-margin hyperplane and the linear classifier it defines is known as a maximum margin classifier, or - equivalently - the perceptron of optimal stability.

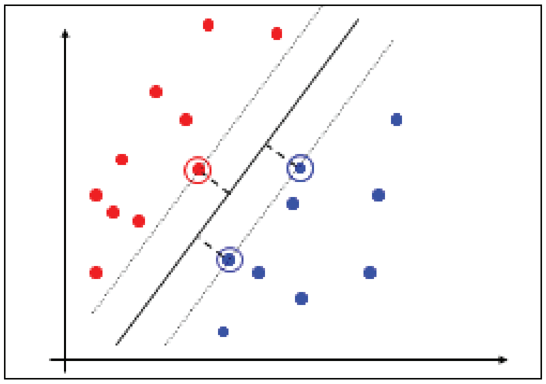


Figure 2: Maximum-Margin Classification

Vapnik has proved the following:

The class of optimal seperators has the VC bounded from above as $H \leq \min\{[D^2/\rho^2], m_0\}+1$

Where ρ is the margin, D is the diameter of the smallest sphere that can enclose all of the training examples and m_0 is the dimensionality. Intuitively, this implies that regardless of dimension m_0 we can minimize the dimension by maximizing the margin ρ. Thus the complexity of the classifier is kept small regardless of dimensionality.

The VC dimension of a set of functions is p if and only if there exists a set of points {xi}p i=1 such that these points can be separated in all 2p possible configurations, and that no set {xi}q i=1 exists where q > p satisfying this property.

WITHIN CLASS AND BETWEEN CLASS SCATTER MATRIX FOR SVM-LDA

LDA or Fisherfaces method overcomes the limitations of eigenfaces method by applying the Fisher's linear discriminant criterion. This criterion tries to maximize the ratio of the determinant of the between-class scatter matrix of the projected samples to the determinant of the within-class scatter matrix of the projected samples.

Fisher discriminants group images of the same class and separates images of different classes. Images are projected from N^2-dimensional space to C dimensional space (where C is the number of classes of images). For example, consider two sets of points in 2-dimensional space that are projected onto a single line.

Depending on the direction of the line, the points can either be mixed together or separated. Fisher discriminants find the line that best separates the points. To identify an input test image, the projected test image is compared to each projected training image, and the test image is identified as the closest training image. As with eigenspace projection, training images are projected into a subspace. The test images are projected into the same subspace and identified using a similarity measure.

The LDA method tries to find the subspace that best discriminates different classes. The within-class scatter matrix, also called intra-personal, represents variations in appearance of the same individual due to different lighting and face expression, while the between-class scatter matrix, also called the extra-personal, represents variations in appearance due to a difference in identity. By applying this method, maximizing the between-class scatter matrix S_b, while minimizing the within-class scatter matrix Sw in the projective subspace.

The within class scatter matrix represents how classes are distributed closely within classes.

$$S_w = \frac{1}{N} \sum_{i=1}^{k} \sum_{x} \sum_{x^o \xi \kappa} (x-m_k)(x-m_k)^T$$

The between class scatter matrix describes how classes are separated from each other.

$$S_b = \frac{1}{N} \sum_{i=1}^{k} N_k\,(m_k - m)(m_k - m)^T$$

X_i denotes the input.
K is the number of clases.
N_k is the sample size of K^{th} class
Mean vector m is $M=\frac{1}{N}\sum_{i=1}^{N} Xi$

LDA considers maximizing the following objectives: $J(w)=\arg_w \min(w^T S_w w / w^T S_b w)$

III. SVM WITH KDA

In this module, we implement SVM classification on the dataset regressed from KDA. KDA is a combination of LDA and its kernel version. KDA accepts both linear and non-linear data.

In this module, we implement SVM classification on the dataset regressed from LDA. Classifying data is a common task in machine learning. Suppose some given data points each belong to one of two classes, and the goal is to decide which class a new data point will be in. In the case of SVMs, a data point is viewed as a p-dimensional vector (a list of p numbers), and we want to know whether we can separate such points with a $(p-1)$ -dimensional hyperplane. This is called a linear classifier. There are many hyperplanes that might classify the data. One reasonable choice as the best hyperplane is the one that represents the largest separation, or margin, between the two classes. So we choose the hyperplane so that the distance from it to the nearest data point on each side is maximized. If such a hyperplane exists, it is known as the maximum-margin hyperplane and the linear classifier it defines is known as a maximum margin classifier, or - equivalently - the perceptron of optimal stability.

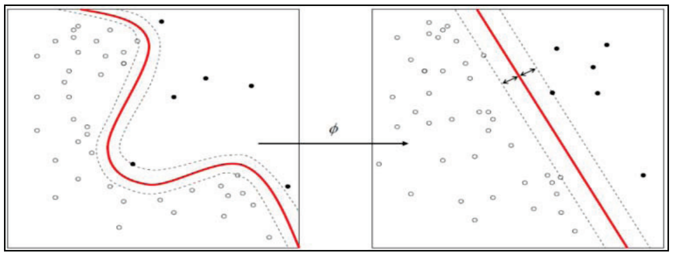


Figure 3: Non-Linear Regression Function

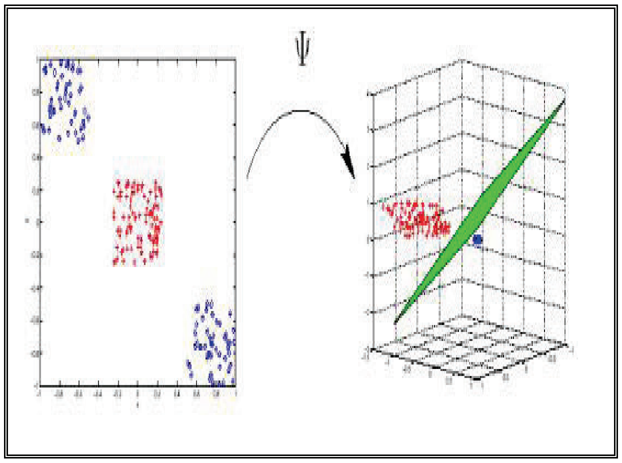


Figure 4: Non-linear mapping of input examples into high dimensional feature space

WITHIN CLASS AND BETWEEN CLASS SCATTER MATRIX FOR SVM- KDA:

For finding the within class and between class matxix for SVM-KDA it will be same as SVM-LDA. By applying this method, maximizing the between-class scatter matrix S_b, while minimizing the within-class scatter matrix Sw in the projective subspace.

The within class scatter matrix represents how classes are distributed closely within classes.

$$S_w = \frac{1}{N} \sum_{i=1}^{k} \sum_{x} \sum_{x^o \xi K} (x-m_k)(x-m_k)^T$$

The between class scatter matrix describes how classes are separated from each other.

$$S_b = \frac{1}{N} \sum_{i=1}^{k} N_k (m_k - m)(m_k - m)^T$$

KDA considers maximizing the following objectives: $J(w)=\arg_w \min(w^T S_w w / w^T S_b w)$

IV. KPCA WITH LDA

Linear combinations are particular attractive because they are simple to compute and analytically tractable. In effect, linear methods project the high-dimensional data onto a lower dimensional subspace.

A data set of data vectors in an m-dimensional space is denoted as

$$A=[a_1,\ldots\ldots\ldots,a_n]=[A_1,\ldots\ldots\ldots,A_r]\epsilon\, R^{m\times n}$$

where tha data is clustered to r classes and each block $A_i \epsilon R^{m\times n}{}_i$ has n_i data vectors. Let N_i $(1 \le i \le r)$ be the set of column indices that belong to the class i. The between-class scatter matrix S_b and the within-class scatter matrix S_w are defined as:

$S_b = \sum_{i=1}^{r} n_i (c_i - c)(c_i - c)^T$ and $S_w = \sum_{i=1}^{r} \sum_{j \in N_i} (a_j - c_i)(a_j - c_i)^T$

Where $c_i = \frac{1}{n_i} \sum_{j \in N_i} a_j$ and

$$c = \frac{1}{n} \sum_{j=1}^{n} a_j$$

are the centroid of the class i and the global centroid, respectively. The separability of classes in a data set can be measured by using the traces of these scatter matrices.

The goal of Linear Discriminant Analysis (LDA) is to find a transformation matrix $G \in R^{m\times l}$ for some integer l with $l \le m$ that defines a linear transformation

$G^T: a \in R^{m\times 1} \rightarrow y = G^T a \in R^{l\times 1}$

and preserves the cluster structure by maximizing the between-class scatter and minimizing the within-class scatter. In the transformed space by G^T , the between-class scatter matrix $\overline{S_b}$ and the within-class scatter matrix $\overline{S_w}$ become

$\overline{S_b} = G^T S_b G$ and $\overline{S_w} = G^T S_w G$

respectively. A commonly used criterion in LDA for finding an optimal clustered structure preserving transformation G^T is

$\max_G trace\ ((G^T S_w G)^{-1}(G^T S_b G))$

It is well known that this criterion is satisfied when l = r -1 where r is the number of the classes in the data, and the columns of $G^T \in R^{m\times(r-1)}$ are the eigenvectors corresponding to the r-1 largest eigenvalues for the eigen

value problem.

$S_w^{-1} S_b x = \lambda x$

However, as in many applications such as information retrieval and face recognition, when the number of data items is smaller than the dimension of data space, S_w becomes singular. Recently , a method which applies the GSVD to solve the generalized eigenvalue problem.

$S_b x = \lambda S_w x$

V. KPCA WITH KDA

In this section, we present a nonlinear extension of LDA based on kernel functions. The main idea of the kernel method is that without knowing the nonlinear feature mapping or the mapped feature space explicitly, we can work on the feature space through kernel functions, as long as the problem formulation depends only on the inner products between data points. This is based on the fact that for any kernel functions k satisfying Mercer's condition, there exists a mapping ϕ such that

$$< \Phi(a),\Phi(b) > = k(a,b)$$

Where <, > is an inner product in the feature space transformed by ϕ. For a finite data set $\{a_1,\ldots a_n\}$, a kernel function k satisfying Mercer's condition can be rephrased as the kernel matrix $K=[k(a_i,a_j)]_{1\le i,j\le n}$ being positive semi-definite. The polynomial kernel

$$K(x,y)=(\gamma_1(x.y) + \gamma_2)^d, d > 0 \text{ and } \gamma_1,\gamma_2 \in R$$

and the Gaussian kernel function:

$$k(x,y) = \exp(-\frac{||X-Y||^2}{2\sigma^2}), \sigma \in R$$

are two of the most widely used kernel functions. The feature map ϕ can be either linear or nonlinear depending on kernel functions used. If the inner product kernel function k(x,y) = x.y is used, the feature map is an identity map. In the kernel methods neither the feature map nor the feature space needs to be formed explicitly due to the relation once the kernel function k is known. We apply the kernel method to perform LDA in the feature space instead of the original input space. Given a kernel function k, let Φ be a mapping satisfying and define $F \in R^n$ to be the

feature space from the mapping Scatter Φ matrices S_b and S_w in feature space F can be expressed as

$S_b = H_b H_b^T$ and $S_w = H_w H_w^T$ where

$$H_b = [\sqrt{n_1}(\overline{c_1} - \overline{c}), \ldots, \sqrt{n_r}(\overline{c_r} - \overline{c})] \in R^{n_i\times 1}$$

$H_w = [\Phi(A_1) - \overline{c_1} e_1^T, \ldots, \Phi(A_r) - \overline{c_r} e_r^T] \in R^{N\times n}$

$$\bar{c}_i = \frac{1}{n_i}\sum_{j\in N_i}\Phi(a_j), \bar{c} = \frac{1}{n}\sum_{i=1}^{n}\Phi(a_i) \text{ and } e_i = [1, \ldots 1]^T \in R^{n_i \times 1}$$

The notations $\Phi(A_i)$ are used to denote $\Phi([a_j,\ldots,a_k]) = [\Phi(a_j),..,\Phi(a_k)]$. Then the LDA in F finds a transformation matrix.

$$\mathbf{g=[\varphi_1,\ldots\varphi_{r-1}]\in R_{N\times(r-1)}}$$

where the columns of g are the generalized eigenvectors corresponding to the r-1 largest eigenvalues of $\mathbf{S_{b\varphi} = \lambda S_{w\varphi}}$

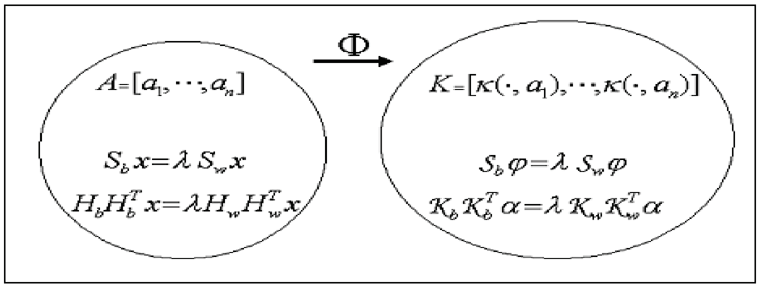
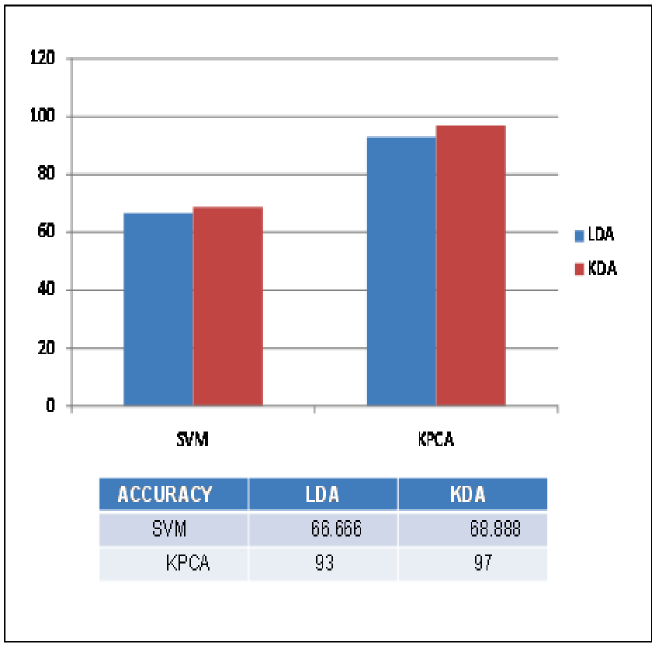

Φ
$A=[a_1,\cdots,a_n]$
$S_b x=\lambda S_w x$
$H_b H_b^T x=\lambda H_w H_w^T x$
$K=[\kappa(\cdot,a_1),\cdots,\kappa(\cdot,a_n)]$
$S_b\varphi=\lambda S_w\varphi$
$K_b K_b^T\alpha=\lambda K_w K_w^T\alpha$

Figure 4: The problem formulations of LDA in the original data space and the feature space defined by a nonlinear mappingϕ through a kernel function, < ϕ(ai),ϕ(aj)> = k(ai, aj). In the kernel matrix K, k(.,ai) denotes a column vector [k(a1,ai),… k(an,ai)]T

VI COMPARISON BETWEEN SVM AND KPCA

The accuracy of both algorithms SVM and KPCA with LDA and KDA are compared and the graph is as follows



ACCURACY	LDA	KDA
SVM	66.666	68.888
KPCA	93	97

Figure 4: Comparison of the graph of SVM and KPCA with LDA and KDA

Table: Results for the Comparison of SVM and KPCA accuracy with LDA and KDA

ACCURACY	LDA	KDA
SVM	66.666	68.888
KPCA	93	97

VII. CONCLUSION

When SVM-LDA is compared with KPCA-LDA, KPCA-KDA gives good accuracy. When SVM-KDA is compared with KPCA-KDA, KPCA-KDA gives good accuracy. Finally KPCA-KDA gives the best accuracy of all. In KPCA we use Gaussian kernal function. KPCA can also be applied to linear, polynomial and sigmoid kernel function for future work.

REFERENCES

[1] Bing-Yu Sun, Jiuyong Li, Desheng Dash Wu, Xiao-Ming Zhang, and Wen-Bo Li "Kernel Discriminant Learning for Ordinal Regression" IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 22, NO. 6, JUNE 2010.

[2] R.H. Erbrich, T. Graepel, and K. Obermayer, "Large Margin Rank Boundaries for Ordinal Regression," Advances in Large Margin Classifiers, pp. 115-132, MIT Press, 2000.

[3] K. Crammer and Y. Singer, "Pranking with Ranking," Advances in Neural Information Processing Systems, T.G. Dietterich, S. Becker, and Z. Ghahramani, eds., vol. 1, no. 14, pp. 641-647, MIT Press, 2002.

[4] A. Shashua and A. Levin, "Ranking with Large Margin Principle: Two Approaches," Advances in Neural Information Processing Systems, vol. 15, pp. 961-968, MIT Press, 2003.

[5] Machine Learning (ICML '05), pp. 145-152, 2005.

[6] L. Lin and H.-T. Lin, "Ordinal Regression by Extended Binary Classification," Advances in Neural Information Processing Systems, vol. 19, pp. 865-872, MIT Press, 2007.

[7] C.M. Bishop, Pattern Recognition and Machine Learning. Springer, 2006.

[8] R.O. Duda, P.E. Hart, and D. Stork, Pattern Classification. Wiley, 2000.

[9] S. Mika, "Kernel Fisher Discriminants," PhD thesis, Univ. of Technology, 2002.

[10] V. Vapnik, The Nature of Statistical Learning Theory. Wiley, 1998.

[11] Jian Yang, Alejandro F. Frangi, Jing-yu Yang, David Zhang, Senior Member, IEEE and Zhong Jin," KPCA Plus LDA: A Complete Kernel Fisher Discriminant Framework for Feature Extraction and Recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 2, February 2005.

[12] S. Kramer, G. Widmer, B. Pfahringer, and M. De Groeve, "Prediction of Ordinal Classes Using Regression Trees," Fundamenta Informaticae, vol. 47,nos. 1/2, pp. 1-13, 2001.

[13] J. Lu, K.N. Plataniotis, and A.N. Venetsanopoulos, "Face Recognition Using Kernel Direct Discriminant Analysis Algorithms," IEEE Trans. Neural Networks, vol. 14, no. 1, pp. 117-126, Jan. 2003.

[14] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin, "Graph Embedding and Extensions: A General Framework for Dimensionality Reduction," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 29, no. 1, pp. 40-51,Jan. 2007.