International Journal of Multimedia and Ubiquitous Engineering
Vol.10, No.9 (2015), pp.91-98
http://dx.doi.org/10.14257/ijmue.2015.10.9.10

Robust Scalable Algorithm Applied to Real World Problem

Byung Joo Kim

Youngsan University Department of Computer Engineering, Korea
bjkim@ysu.ac.kr

Abstract

Advances in digital sensors, communications, computation, and storage have created huge collections of data, capturing information of value to business, science, government, and society. Many machine learning algorithms do not scale beyond data sets of a few million elements or cannot tolerate the statistical noise and gaps found in real-world data. Further research is required to develop algorithms that apply in real-world situations and on data sets of trillions of elements. In this paper we propose a conjugate based novel algorithm to handle the huge collections of data. Through the experimental results, proposed method performs well on huge data from UCI machine learning repository data Set.

Keywords: *Conjugate Method, LS-SVM, Support Vector Machine*

1. Introduction

It is now a recognized fact that we are facing a data revolution in both sciences and industry, giving rise to databases of unprecedented scale (*e.g.*, distributed databases or data streams), as well as altogether new data formats (*e.g.*, free-form text, networks, *etc.*). The availability of big data presents an unprecedented opportunity, but also an unprecedented challenge. The machine learning and big data group are rising to this challenge by developing machine learning techniques that can handle modern data types, and draw on statistical and computational intelligence techniques to navigate vast amounts of information with minimal human supervision. Traditional machine learning has been largely concerned with developing techniques for small or modestly sized datasets. These techniques fail to scale up well for large data, a situation becoming increasingly common in today's world. Furthermore most of the machine learning classifiers are trained in a batch way. Under this model, all training data is given a priori and training is performed in one batch. If more training data is later obtained the classifier must be re-trained from scratch. Re-solving the problem from scratch seems computationally wasteful. In this research we will focus on developing classifier for big data sets and incremental way of learning for dealing with real world problem. Recently kernel trick has been applied to PCA and is based on a formulation of PCA in terms of the dot product matrix instead of the co-variance matrix [1]. Kernel PCA (KPCA), however, requires storing and finding the eigenvectors of a N × N kernel matrix where N is a number of patterns. It is infeasible method when N is large. This fact has motivated the development of incremental way of KPCA method which does not store the kernel matrix. It is hoped that the distribution of the extracted features in the feature space has a simple distribution so that a classifier could do a proper task. But it is point out that extracted features by KPCA are global features for all input data and thus may not be optimal for discriminating one class from others [2]. This has naturally motivated to combine the feature extraction method with classifier for classification purpose. In this paper we propose a new classifier for on-line and big data. Paper is composed of as follows. In Section 2 KPCA and eigen space update

ISSN: 1975-0080 IJMUE

criterion is introduced. Conjugate based LS-SVM method is described in Section 3. Experimental results to evaluate the performance of proposed classifier is shown in Section 4. Discussion of proposed classifier and future work is described in Section 5.

2. Incremental KPCA

In this Section, we will give a brief introduction to the method of incremental PCA algorithm which overcomes the computational complexity of standard PCA. Before continuing, a note on notation is in order. Vectors are columns, and the size of a vector, or matrix, where it is important, is denoted with subscripts. Particular column vectors within a matrix are denoted with a superscript, while a superscript on a vector denotes a particular observation from a set of observations, so we treat observations as column vectors of a matrix. As an example,

A^{i}_{mn} is the ith column vector in an $m \times n$ matrix. We denote a column extension to a matrix using square brackets. Thus $[A_{mn} b]$ is an(m × (n + 1)) matrix, with vector b appended to A_{mn} as a last column.

To explain the incremental PCA, we assume that we have already built a set of eigenvectors $U = [u_j, j = 1, \cdots, k]$ after having trained the input images $x_i, i =, \cdots, N$. The corresponding eigenvalues are Λ and $\overline{\text{X}}$ is the mean of input image. Incremental building of eigenspace requires to update these eigenspace to take into account of a new input image. Here we give a brief summarization of the method which is described in [5]. First, we update the mean:

$$\overline{x}' = \frac{1}{N+1}(N\overline{x} + x_{N+1}) \tag{1}$$

We then update the set of eigenvectors to reflect the new input image and to apply a rotational transformation to U. For doing this, it is necessary to compute the orthogonal residual vector $\hat{h} = (Ua_{N+1} + \overline{x}) - x_{N+1}$ where

a_{N+1} is principal component and normalize it to obtain $h_{N+1} = \frac{h_{N+1}}{\left\|h_{N+1}\right\|_2}$ for $\left\|h_{N+1}\right\|_2 \rangle 0$ and $h_{N+1} = 0$ otherwise. We obtain the new matrix of eigenvectors U' by appending h_{N+1} to the eigenvectors U and rotating them:

$$U' = [U, h_{N+1}]R \tag{2}$$

Where R∈ $\Re^{(k+1)\times(k+1)}$ is a rotation matrix. R is the solution of the eigenspace of the following form:

$$DR = R\Lambda' \tag{3}$$

Where Λ' is a diagonal matrix of new eigenvalues. We compose D $\in \Re^{(k+1)\times(k+1)}$ as:

$$D = \frac{N}{N+1}\begin{bmatrix} \Lambda & 0 \\ 0^T & 0 \end{bmatrix} + \frac{N}{(N+1)^2}\begin{bmatrix} aa^T & \gamma a \\ \gamma a^T & \gamma^2 \end{bmatrix} \tag{4}$$

Where $\gamma = h^T_{N+1}(x_{N+1} - \overline{x})$ and $a = U^T(x_{N+1} - \overline{x})$. Though there are other ways to construct matrix D [4][5], the only method ,however, described in [6] allows for the updating of mean.

2.1. Eigenspace Updating Criterion

The incremental method should include an additional eigenvector if necessary. In our previous research we can't set explicit rule for adding a eigenvector. In this Section we will give a guide line for this problem. The incremental PCA represents the input data with principal components $a_{i(N)}$ and it can be approximated as follows:

$$\hat{x}_{i(N)} = Ua_{i(N)} + \bar{x} \tag{5}$$

To update the principal components $a_{i(N)}$ for a new input x_{N+1}, computing an auxiliary vector η is necessary. η is calculated as follows:

$$\eta = [U \ \hat{h}_{N+1}]^{T} (\bar{x} - \bar{x}') \tag{6}$$

Then the computation of all principal components is

$$a_{i(N+1)} = (R')^{T} \begin{bmatrix} a_{i(N)} \\ 0 \end{bmatrix} + \eta, \quad i = 1, \cdots, N+1 \tag{7}$$

The above transformation produces a representation with (k + 1) dimensions. Due to the increase of the dimensionality by one, however, more storage is required to represent the data. If we try to keep a k dimensional eigenspace, we lose a certain amount of information. It is needed for us to set the criterion on retaining the number of eigenvectors. There is no explicit guideline for retaining a number of eigenvectors. Here we introduce some general criteria to deal with the model's dimensionality:

(a) Adding a new vector whenever the size of the residual vector exceeds an absolute threshold.

(b) Adding a new vector when the percentage of energy carried by the last eigenvalue in the total energy of the system exceeds an absolute threshold, or equivalently, defining a percentage of the total energy of the system that will be kept in each update.

(c) Discarding eigenvectors whose eigenvalues are smaller than a percentage of the first eigenvalue.

(d) Keeping the dimensionality constant.

In this paper we take a rule described in (b). We set our criterion on adding an eigenvector as $\lambda'_{k+1} \rangle 0.7\bar{\lambda}$ where $\bar{\lambda}$ is a mean of the λ. Based on this rule, we decide whether adding u'_{k+1} or not.

Incremental Kernel PCA Algorithm

```
$X = (x^1 x^2 \ldots x^n)$   : matrix of training examples:
$\lambda$ : initial eigenvalue, U : initial eigenvector
$K(x, y) := \Phi(x)' \Phi(y)$   :  initial kernel matrix ,
$\bar{x}$  : initial mean

for k=1:n                    : begin re-learning iteration
    $\bar{x}' = \frac{1}{n+1}(n\bar{x} + x_{n+1})$  : update the mean
    $h_{n+1} = (Ua_{n+1} + \bar{x}) - x_{n+1}$  : compute orthogonal residual vector
    $\hat{h}_{n+1} = \frac{h_{n+1}}{\|h_{n+1}\|_2}$  for  $\|h_{n+1}\|_2 > 0$ ,  $\hat{h}_{n+1} = 0$        otherwise
    $D = \frac{n}{n+1}\begin{bmatrix} \Lambda & 0 \\ 0^T & 0 \end{bmatrix} + \frac{n}{(n+1)^2}\begin{bmatrix} aa^T & \gamma a \\ \gamma a^T & \gamma^2 \end{bmatrix}$  :Construct matrix D
    where  $\gamma = \hat{h}_{n+1}^T (x_{n+1} - \bar{x})$ ,  $a = U^T (x_{n+1} - \bar{x})$
    $DR = R\Lambda'$  : solve the eigenproblem i.e computerotation matrix R
    If   ((n+1) $\lambda_{N+1}$   > 0.7  $\lambda$)
        Update eigenvector using the criterion rule
    else
        retain current eigenspce
end for                : end of re-learning iteration
```

3. Conjugate LS-SVM for Real World Data

Support vector machines (SVM) developed by Vapnik [7] and it is a powerful methodology for solving problems in nonlinear classification. Originally, it has been introduced within the context of statistical learning theory and structural risk minimization. In the methods one solves convex optimization problems, typically by quadratic programming (QP). Solving QP problem requires complicated computational effort and need more memory requirement. LS-SVM [8] overcomes this problem by solving a set of linear equations in the problem formulation. LS-SVM method is computationally attractive and easier to extend than SVM. But traditional batch way LS-SVM requires storing (N+1) × (N+1) matrix where N is a number of patterns. It is infeasible method when dealing with big data. For big data sets the use of iterative methods is recommended. In principle, various methods can be used at this point including SOR (Successive Over-Relaxation), CG (Conjugate Gradient), GMRES (Generalized Minimal Residual) *etc.* However, not all of these iterative methods can be applied to any kind of linear system. For example, in order to apply CG the matrix should be positive definite. Due to the presence of the b bias term in the LS-SVM model the resulting matrix is not positive definite. So before we can apply such methods we have to transform the linear system into a positive definite system. The LS-SVM KKT system is of the form

$$\begin{bmatrix}0 & Y^T\\ Y & H\end{bmatrix}\begin{bmatrix}\xi_1\\ \xi_2\end{bmatrix} = \begin{bmatrix}d_1\\ d_2\end{bmatrix} \tag{8}$$

More specifically with $H = \Omega + I/\gamma$, $\xi_1 = b, \xi_2 = \alpha$, $d_1 = 0, d_2 = I_v$. This can be transformed into

$$\begin{bmatrix}s & 0\\ 0 & H\end{bmatrix}\begin{bmatrix}\xi_1\\ \xi_2 + H^{-1}y\xi_1\end{bmatrix} = \begin{bmatrix}-d_1 + y^TH^{-1}d_2\\ d_2\end{bmatrix} \tag{9}$$

With $S = y^TH^{-1}y > 0$ ($H = H^{-T} > 0$) Because s is positive and H positive definite the overall matrix is positive definite. This form is very suitable because different kinds of iterative methods can be applied to problems involving positive definite matrices. This leads to the LS-SVM classifier with conjugate gradient algorithm LS-SVM for big data is as follows

1. Solve η,ν from $H\eta = Y$ and
   $H\nu = 1_v$
2. Compute $s = Y^T \eta$
3. Find solution
   $b = \eta^T 1_v / s$
   $\alpha = \nu - \eta b$

4. Experiment

To evaluate the performance of proposed classification system, experiment is performed on big data. First we evaluate the proposed system to HIV-1 protease cleavage data set.

4.1. HIV-1 protease cleavage Data Set

HIV-1 protease cleavage data set [11] contains 6590 instances and number of attributes are 1. Details on references where data has been collected are provided for the 746 and the 1625 data set. The origins of the Schilling data and the Impens data are described in the paper [12]. The 746 and 1625 data sets share many patterns. The first 80% are provided here as the training dataset and the remaining 20% as the testing dataset.

Table 1. Training and Generalization Result on HIV-1 Protease Cleavage Data

	Training	Generalization	Eigenvalue update criterion
Standard LS-SVM	100%	98. 2%	none
Proposed method	100%	97.8%	$\lambda' \rangle 0.7\bar{\lambda}$

In [10] it is shown that the use of 10-fold cross-validation for hyperparameter selection of LS-SVMs consistently leads to very good results. In this problem RBF kernel has been used.

Table 1. Training and Generalization Result on HIV-1 Protease Cleavage Data

	Training	Generalization	Eigenvalue update criterion
Standard LS-SVM	100%	98.7%	none
Proposed method	100%	97.1%	$\lambda^{'} \rangle 0.7\bar{\lambda}$

The results on the HIV-1 protease cleavage data set are given in Table 1. Generalization ability in proposed method is similar to standard LS-SVM. But in standard LS-SVM (1176,531) x (1176,531) matrix is needed. It is not infeasible for big data.

4.2. TV News Channel Commercial Detection Data Set

TV news channel commercial detection data set [13] contains 129685 instances and number of attributes are 12. Automatic identification of commercial blocks in news videos finds a lot of applications in the domain of television broadcast analysis and monitoring. Commercials occupy almost 40-60% of total air time. Manual segmentation of commercials from thousands of TV news channels is time consuming, and economically infeasible hence prompts the need for machine learning based Method. Classifying TV News commercials is a semantic video classification problem. TV News commercials on particular news channel are combinations of video shots uniquely characterized by audio-visual presentation. Hence various audio visual features extracted from video shots are widely used for TV commercial classification. Indian News channels do not follow any particular news presentation format, have large variability and dynamic nature presenting a challenging machine learning problem. Features from 150 Hours of broadcast news videos from 5 different (3 Indian and 2 International News channels) news channels. Viz. CNNIBN, NDTV 24X7, TIMESNOW, BBC and CNN are presented in this dataset. Videos are recorded at resolution of 720 X 576 at 25 fps using a DVR and set top box. 3 Indian channels are recorded concurrently while 2 International are recorded together. Feature file preserves the order of occurrence of shots. The first 80% are provided here as the training dataset and the remaining 20% as the testing dataset.

Table 2. Training and Generalization Result on TV News Channel Data Set

	Training	Generalization	Eigenvalue update criterion
Standard LS-SVM	100%	97. 7%	none
Proposed method	100%	96.9%	$\lambda^{'} \rangle 0.7\bar{\lambda}$

The results on the TV news channel commercial detection data are given in Table 2. Generalization ability in proposed method is similar to standard LS-SVM. But in standard LS-SVM (2144,724) x (2144724) matrix is needed. It is not infeasible for big data.

4.3. Gas Sensor Array under Dynamic Gas Mixtures Data Set

Gas sensor array under dynamic gas mixtures data set [14] contains 4178504 instances and number of attributes are 19. The data set was collected in a gas delivery platform facility at the ChemoSignals Laboratory in the BioCircuits Institute, University of California San Diego. The measurement system platform provides versatility for obtaining the desired concentrations of the chemical substances of interest with high accuracy and in a highly reproducible manner. The sensor array included 16 chemical sensors (Figaro Inc., US) of 4 different types: TGS-2600, TGS-2602, TGS-2610, TGS-2620 (4 units of each type). The sensors were integrated with customized signal conditioning and control electronics. The operating voltage of the sensors, which controls the sensors operating temperature, was kept constant at 5 V for the whole duration of the experiments. The sensorsâ€™ conductivities were acquired continuously at a sampling frequency of 100 Hz. The sensor array was placed in a 60 ml measurement chamber, where the gas sample was injected at a constant flow of 300 ml/min. Each measurement was constructed by the continuous acquisition of the 16-sensor array signals while concentration levels changed randomly. For each measurement (each gas mixture), the signals were acquired continuously for about 12 hours without interruption. Training and test dataset ratio is the same as above experiment

Table 3. Training and Generalization Result on Gas Sensor Data Set

	Training	**Generalization**	**Eigenvalue update criterion**
Standard LS-SVM	100%	98.02%	None
Proposed method	100%	97.8%	$\lambda^{'} \rangle 0.7\overline{\lambda}$

The results on the Gas sensor array under dynamic gas mixtures data set are given in Table 3. Generalization ability in proposed method is similar to standard LS-SVM. But in standard LS-SVM (5,632,843) x (5,632,843) matrix is needed. It is not infeasible for big data.

5. Conclusion and Remarks

A conjugate based LS-SVM which combining incremental KPCA was presented for dealing with big data. Such classifier has following advantages. Proposed classifier is more efficient in memory requirement than batch LS-SVM. In batch LS-SVM the (N+1) × (N+1) matrix has to be stored, while for our proposed method does not. It is very useful when dealing with big data. Experimental results on huge data from UCI machine learning repository, proposed method shows lead to good performance.

Acknowledgment

This study was supported by a grant of Youngsan University(in 2014).

References

[1] V. N. Vapnik, “Statistical learning theory. John Wiley & Sons”, New York, **(1998)**.

[2] H. Gupta, A. K. Agrawal, T. Pruthi, C. Shekhar and R. Chellappa, “An Experimental Evaluation of Linear and Kernel-Based Methods for Face Recognition”, accessible at http://citeseer.nj.nec.com.

[3] P. Hall, D. Marshall and R. Martin, “Incremental eigenalysis for classification”, In British Machine Vision Conference, vol. 1, September **(1998)**.

[4] J. Winkeler, B. S. Manjunath and S. Chandrasekaran, “IEEE Computer Society Press. Subset selection for active object recognition”, **(1999)**.

[5] H. Murakami and B. V. K.V. Kumar, “IEEE PAMI, Efficient calculation of primary images from a set of images”, vol. 4, no. 5, **(1982)**.

[6] B. J. Kim, J. Y. Shim, C. H. Hwang and I. K. Kim, “Incremental Feature Extraction Based on Emperical Feature Map”, Lecture Notes in Artificial Intelligence, vol. 2871, **(2003)**.

[7] V. N. Vapnik, “Statistical learning theory. John Wiley & Sons”, New York, **(1998)**.

[8] A. K. Suykens and J. Vandewalle, “Neural Processing Letters”, Least squares support vector machine classifiers, vol. 9, **(1999)**.

[9] G. H. Golub and C. F. Van, “Large scale LS_SVM Matrix Computations”, Baltimore MD Johns Hopkins University, **(2003)**.

[10] V. Gestel, T. J. A. K. Suykens, G. Lanckriet, Lambrechts A.B. De Moor and J. Vandewalle, "A Bayesian Framework for Least Squares Support Vector Machine Classifiers", Internal Report 00-65, ESAT-SISTA, K.U. Leuven, **(2001)**.
[11] http://archive.ics.uc*i.e*du/ml/datasets/HIV-1+protease+cleavage /.
[12] T. Rögnvaldsson, L. You and D. Garwicz, "Bioinformatics State of the art prediction of HIV-1 protease cleavage sites", **(2015)**.
[13] http://archive.ics.uc*i.e*du/ml/datasets/TV+News+Channel+Commercial+Detection+Dataset
[14] http://archive.ics.uc*i.e*du/ml/datasets/Gas+sensor+array+under+dynamic+gas+mixtures