Available online at www.elixirpublishers.com (Elixir International Journal)

Computer Science and Engineering

Elixir Comp. Sci. & Engg. 43 (2012) 6755-6758

ISSN: 2229-712X

Survey on feature selection methods in image information mining

R.Maruthamuthu[1], A.John Sanjeev Kumar[2] and S.Ugendran[1]
[1]M.Sc Software Engineering RVS College of Engineering and Technology, Dindigul.
[2]Department of Computer Applications, Thiagarajar College of Engineering, Madurai.

ARTICLE INFO

Article history:
Received: 7 December 2011;
Received in revised form:
25 January 2012;
Accepted: 7 February 2012;

Keywords
Feature selection,
Image information mining (IIM), Principal Component Analysis (PCA) & Independent Component Analysis (ICA).

ABSTRACT
Image information mining (IIM) approaches produce enormous amounts of features that are computationally expensive and in- efficient to process before the actual information discovery takes place[1]. Also, it is complicated because the combination of the features has little relevance to the hypothesis space. Hence, selecting a relevant subset of features is necessary to overcome these problems and to provide an efficient representation of the target class. In this paper, we propose survey on feature selection and feature transformations.

Introduction

In a coastal disaster event (e.g., hurricanes, flooding), it is necessary to have access to the information on damaged areas in near real time for effective deployment of rescue and recovery measures. The dissemination of information that time critical calls for systems that will facilitate quick assessment of the scenario from multiple perspectives. The problem is that new classes are created in disasters that were not used in the image information mining (IIM) scheme of interest. For example, classes may have been learned dealing with roadways, intact buildings, and forests. However, after a disaster, it is now necessary to classify flooded roadways, buildings with no roofs and knocked down forests from limited training data. Part of the confusion stems from the number of factors that have to be controlled. The performance of PCA depends on the task statement, the subspace distance metric, and the number of subspace dimensions retained. The performance of ICA depends on the task, the algorithm used to approximate ICA, and the number of subspace dimensions retained.

Even more confusingly, there are two very different applications of ICA to face recognition. ICA can be applied so as to treat images as random variables and pixels as observations, or to treat pixels as random variables and images as observations. In keeping with , we refer to these two alternatives as ICA architecture I and architecture II, respectively. There is therefore a space of possible PCA/ICA comparisons, depending on at least five factors. This paper explores this space, in order to find the best technique for recognizing
(1) Subject identity and
(2) Facial actions in face images.

Another reason to explore the space of PCA/ICA comparisons is to provide data for the current debate over global versus local features. The basis vectors that define any subspace can be thought of as image features. Viewed this way, PCA and ICA architecture II produce global features, in the sense that every image feature is influenced by every pixel. (Equivalently, the basis vectors contain very few zeroes.) Depending on your preference, this makes them either susceptible to occlusions and local distortions, or sensitive to holistic properties. Alternatively, ICA architecture I

Produces spatially localized features that are only influenced by small parts of the image. It has been argued that this will produce better object recognition, since it implements recognition by parts .If localized features are indeed superior, ICA architecture I should outperform PCA and ICA

Architecture II. This paper will show empirically that the choice of subspace projection algorithm depends first and foremost on the nature of the task. Some tasks, such as facial identity recognition, are holistic and do best with glob al feature vectors. Other tasks, such as facial action recognition, are local and do better with localized feature vectors. For both

types of tasks, ICA can outperform PCA, but only if the ICA architecture is selected with respect to the task type (ICA architecture I for localized tasks, ICA architecture II for holistic tasks). Furthermore, performance is optimized

if the ICA algorithm is selected based on the architecture (Info Max for architecture I, Fast ICA for architecture II).

When PCA is used, the choice of subspace distance measure again depends on the task.

On one specific component of the overall IIM process, i.e., feature selection and generation. The following are the specific objectives of this research:
1) reduce the number of features (feature subset selection) using wrapper-based genetic algorithm (GA) approach to build predictive models for each disaster-affected land- cover classes and use the models for rapid classification and retrieval of disaster-affected regions;

Tele:
E-mail addresses: maruthamuthu81@gmail.com

2) explore the ability to generate new features (feature generation) from the already extracted low-level features via the wrapper-based approach and assess whether it improves the classification accuracy;

3) Compare the foregoing results with the classical principal components analysis (PCA) weighting and backward weighting methods.

These earlier efforts in the IIM area were focused mainly on the reduction of features using clustering approaches. The very high number of dimensions of the feature vectors can make IIM systems impractical in real-world scenarios. Several approaches such as K-means/modified-K-means clustering , PCA/ Kernel PCA , soft-clustering approach , etc., are currently being used to overcome this problem. However, little was re- ported on the selection of the best feature subsets in IIM. In our view, this is of more importance than the clustering of the data features as feature-data reduction, irrespective of understanding which features are optimal for the prediction of a particular semantic class or a set of classes, does not permit maximum exploitation of the hypothesis space. Hence, predictive-model development should go in combination with feature selection and feature-generation approaches.

II.Principal Component Analysis(PCA)

PCA is probably the most widely used subspace projection technique for face recognition. PCA basis vectors are computed from a set of training images I. As a first step, the average image in I is computed and subtracted from the training images, creating a set of data samples

$t1,t2,\ldots\ldots tn \in I - \bar{I}$

These data samples are then arrayed in a matrix X, with one column per sample image

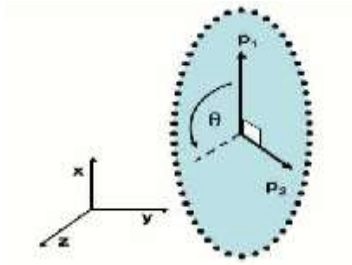
$X=[[i1]\ldots[in]]$

is then the sample covariance matrix for the training images, and the principal $\| \cdots \|$ components of the covariance matrix are computed by solving

$R^T(XX^T)R=\Lambda$

where Λ is the diagonal matrix of eigen values and R is the matrix of ortho normal eigenvectors. Geometrically, R is a rotation matrix2 that rotates the original coordinate system onto the eigenvectors, where the eigenvector associated with the largest eigen value is the axis of maximum variance, the eigenvector associated with the second largest eigen value is the orthogonal axis with the second largest variance, etc. Typically, only the N eigenvectors associated with the largest eigen values are used to define the subspace, where N is the desired subspace dimensionality. There are three related arguments for matching images in the subspace of N eigenvectors. The first is compression. It is computationally more efficient to compare images in subspaces with significantly reduced dimensions. For example, image vectors with 65,536 pixels (256x256) might be projected into a subspace with only 100 to 300 dimensions. The second argument assumes that the data samples are drawn from a normal distribution. In this case, axes of large variance probably correspond to signal, while axes of small variance are probably noise. Eliminating these axes therefore improves the accuracy of matching. The third argument depends on a common preprocessing step, in which the mean value is subtracted from every image and the images are scaled to form unit vectors. This projects the images into a subspace where Euclidean distance is inversely proportional to correlation between the source images. As a result, nearest neighbor matching in eigen space becomes an efficient approximation to image correlation.

This slightly liberal definition of rotation also includes reflection



III. Independent Component Analysis(ICA)

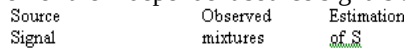
While PCA de correlates the input data using second-order statistics and thereby generates compressed data with minimum mean-squared re projection error, ICA minimizes both second-order and higher-order dependencies in the input. It is intimately related to the blind source separation (BSS) problem, where the goal is to decompose an observed signal into a linear combination of unknown independent signals. Let's be the vector of unknown source signals and x be the vector of observed mixtures. If A is the unknown mixing matrix, then the mixing model is written as

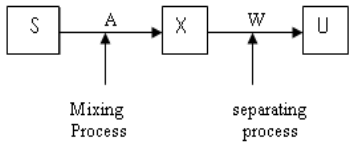
$x=As$

It is assumed that the source signals are independent of each other and the mixing matrix A is invertible. Based on these assumptions and the ob served mixtures, ICA algorithms try to find the mixing matrix A or the separating matrix W such that

$u=Wx=WAs$

is an estimation of the independent source signals .





ICA can be viewed as a generalization of PCA. As previously discussed, PCA de correlates the training data so that the sample covariance of the training data is zero. Whiteness is a stronger constraint that requires both de correlation and unit variance. The whitening transform can be determined as D-1/ 2 RT where D is the diagonal matrix of the eigen values and R is the matrix of orthogonal eigenvectors of the sample covariance matrix. App lying whitening to observed mixtures, however, results in the source signal only up to an orthogonal transformation. ICA goes one step further so that it transforms the whitened data into a set of statistically independent signals .

Signals are statistically independent when

$f_u(u)=\Pi_i f_{ui}(u_i)$_

where fu is the probability density function of u.(It is equivalent to say that the vectors u are uniformly distributed.)

Unfortunately, there may not be any matrix W that fully satisfies the independence condition, and there is no closed form expression to find W. Instead, there are several algorithms that iteratively approximate W so as to indirectly maximize independence. Since it is difficult to maximize the independence condition above directly, all common ICA algorithms recast the problem to iteratively optimize a smooth function whose global optima occurs when the output vectors u are independent. For example, Info Max relies on the observation that independence is maximized when the entropy
H(u) is maximized, where:

$H(u)=-\int f_u(u)\log f_u(u)du$

Info Max performs gradient ascent o n the elements wij so as to maximize H(u) . (It gets its name from the observation that maximizing H(u) also maximizes the mutual information I(u, x) between the input and output vectors.) The JADE algorithm minimizes the kurtosis of fu(u) through a joint diagonalization of the fourth order cumulates, since minimizing kurtosis will also maximize statistical independence. Fast ICA is arguably the most
general, maximizing

$J(y)\approx c[E\{G(y)\}-E\{G(v)\}]^2$

where G is a non-quadratic function, is a Gaussian random variable, and c is any positive constant, since it can be shown that maximizing any function of this form will also maximize independence . Info Max, JADE and Fast ICA all maximize functions with the same global optima . As a result, all three algorithms should converge to the same solution for any given data set. In practice, the different formulations of the independence constraint are designed to enable differentapproximation techniques, and the algorithms find different solutions because of differences among these techniques. Limited empirical studies suggest that the differences in performance between the algorithms are minor and depend on the data set. For example, Zibulevsky and Pearlmutter test all three algorithms on a simulated blind-source separation problem, and report only small differences in the relative error rate: 7.1% for Info Max, 8.6% for Fast ICA, and 8.8% for JADE . On the other hand, Karvanenetal report on another simulated blind-source separation problem where JADE slightly outperforms Fast ICA, with Info Max performing significantly worse. Zieheetal. report no significant difference between Fast ICA and JADE at separating noise from signal in MEG data .In studies using images, Moghaddam and Leeetal report qualitatively similar results for JADE and Fast ICA, but do not publish numbers.

IV.Fast Kernel Independent Component Analysis

This package contains a Mat lab implementation of the Fast Kernel ICA algorithm, as described in Haoshen, Stefanie Jegelka, and Arthur Gretton .The goal of ICA is to separate linearly mixed sources to minimize the statistical dependence between the estimated unmixed sources. Kernel ICA algorithms use kernel measures of statistical independence as their optimization criteria . Fast Kernel ICA (Fast KICA) employs an approximate Newton method to perform the optimization efficiently for larger-scale problems. The kernel independence criterion we use here is the Hilbert-Schmidt norm of the covariance operator in feature space. Another interpretation of this criterion is as a characteristic function based independence measure, as used previously in ICA by CheBic and EriKoi.

The functions 'chol_gauss' and 'amariD' are based on code from Francis Bach (available here). The derivative is computed as described in JegGre (for incomplete Cholesky decomposition).

Code

fastkica.m	main routine
README.txt	Instructions on use
utils\chol_gauss.c	IncompleteCholesky decomposition
utils\dChol2.c	Mex code for derivative .
utils\dChol2Lin.c	Mex code for derivative
utils\Kmn.c, dKmnLin.c	Mex code,
utils\getKern.c	Mex code to compute
utils\compDerivChol.m	Matlabinterface for the gradient.
utils\dChol.m,dCholLin.m	Needed for the gradient
utils\hsicChol.m	Computes HSIC.
utils\hessChol.m	Computes the Hessian.

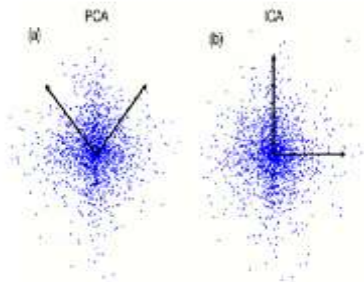
Datasets

The following data and code are also provided in the zip archive, for demonstration purposes.

demo.m Code for demo of kernel ICA.
amariD.m Amari divergence source2.wav Data file
source3.wav Data file
source4.wav Data file

Probe Set	ICA (InfoMax) Arch I		ICA (InfoMax) Arch II		PCA			
	Cosine	L2	Cosine	L2	L1	L2	Cosine	Mahalanobis
fafb (1185)	73.73%	75.96%	82.28%	74.90%	68.42%	72.00%	78.71%	75.28%
fafc (194)	5.67%	5.15%	51.03%	35.57%	10.82%	4.64%	4.64%	39.69%
dup I (722)	35.25%	32.96%	48.48%	37.01%	40.30%	33.34%	35.12%	39.34%
dup II (234)	14.53%	14.53%	12.48%	25.64%	22.22%	14.53%	15.38%	24.39%
Total (2345)	50.82%	50.79%	64.31%	55.31%	57.21%	49.17%	48.03%	56.14%

Table for comparison of PCA and ICA



Conclusion

This paper compares principal component analysis (PCA) and independent component analysis (ICA) in the context of a baseline analysis system, a comparison motivated by contradictory claims in the literature. This paper shows how the relative performance of PCA and ICA depends on the task statement, the ICA architecture, the ICA algorithm, and (for PCA) the subspace distance metric. It then explores the space of PCA/ICA comparisons by systematically testing two ICA algorithms and two ICA architectures against PCA.

References

[1] D. Aha and R. A. Bankert, “Comparative evaluation of sequential featureselection algorithms,” in Proc. Artif Intell. Stat., D. Fisher and J. H. Lenz, Eds., New York, 1996,

[2] M. Bressan, D. Guillamet, and J. Vitria, “Using an ICA representation of high dimensional data for object recognition and classification,” Pattern Recognit, vol. 36, no. 3, pp. 691–701, Mar. 2003.

[3] J. Eriksson,J. Karvanen, and V. Koivunen, "Maximum Likelihood Estimation of ICA-model for Wide Class of Source Distributions," presented at Neural Networks in Signal Processing, Sydney, 2000.

[4] A. Hyvärinen, "The Fixed-point Algorithm and Maximum Likelihood Estimation for Independent Component Analysis," Neural Processing Letters, vol. 10, pp. 1-5, 1999.

[5] A. Hyvärinen and E. Oja, “Independent component analysis: Algorithms and applications,” Neural Netw., vol. 13, no. 4/5, pp. 411–430, May/Jun. 2000.

[6]JonathonShlens”MATLABCodeforPCA”http://www.snl.salk.edu/~shlens/pub/notes/pca.pdf.

[7] A. C. Kak and A. M. Martinez "PCA versus LDA," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, pp. 228-233, 2001.

[8] C. Liu and H. Wechsler, "Comparative Assessment of Independent Component Analysis (ICA) for Face Recognition," presented at International Conference on Audio and Video Based Biometric Person Authentication, Washington, D.C., 1999.

[9] Nicolas H. Younan, Roger L. King and Surya S. Durbha “Wrapper-Based Feature Subset Selection for Rapid Image Information Mining” IEEE Geo Science and Remote Sensing Letters, Vol. 7, No. 1, January 2010.

[10] Nicolas H. Younan, Roger L. King, Surya S. Durbha and Vijay P. Shah ”Feature Identification via a Combined ICA–Wavelet Method for Image Information Mining” IEEE Geo Science and Remote Sensing Letters, Vol. 7, No. 1, January 2010.