Feature Extraction on the AVIRIS dataset

Manuel Montoya Catalá
Master in Multimedia and Communications
Carlos III of Madrid University
Spain, PA 81129
mmontoya@ing.uc3m.es

Abstract

This paper analyses and compares the performance of several feature extraction techniques and classifier systems for the AVIRIS image Indian Pine dataset. Several linear feature extraction methods (PCA, LDA, ICA) are used in order to study their properties and performance for different classier techniques such as LR, LDA and SVM evaluating their advantages.

1 Introduction

In this paper several a study of several feature extraction techniques and classification schemes is developed for the AVIRIS image Indian Pine dataset [1][2]. The goal of this paper is to expose the properties of feature extraction and how it affects to the performance of different classifying systems. Feature extraction can improve a classification system in several ways; first, it can transform the data into a new space that is easier for the classifiers to learn, second, it can perform dimensionality reduction, increasing the tractability of the problem and reducing the curse of dimensionality. A more compact representation of the data reduces the number of parameters of a classifier to be tuned and the time spent in learning. Feature extraction obtains the relevant data from the initial data, removing irrelevant/noisy/correlated components; the loss of relevant information should be minimized and discovering good combinations of input variables.

2 Linear methods

Linear methods generate a new feature space $\bar{\bar{X}}_{new}$ as a linear combination of the initial feature space $\bar{\bar{X}}_{ini}$. Each new feature is obtained by multiplying a sample vector from the initial dataset by the transformation vector associated to the feature. The set of transforming vectors form the transformation matrix $\bar{\bar{U}}$, so that $\bar{\bar{X}}_{new} = \bar{\bar{U}}\bar{\bar{X}}_{ini}$. The goal of a linear method is to find the transformation matrix $\bar{\bar{U}}$ that optimizes some criteria. A linear projection is equal to a rotation of axis in the feature space.

2.1 PCA

PCA is an unsupervised method that finds the projections (hyperplanes) over the dataset that maximizes the variance of the projected data. A training matrix with N samples and D dimensions will form a matrix (NxD), being $N \ll D$, the matrix will have rank N. The N D-divisional vectors form a basis of the subspace where the images are. PCA obtains a new set of basis vectors which are orthogonal and whose projections have maximum variance, Figure 1 (a) shows the 2 projection vectors of a reduced version of the dataset in which we only take the 2 first features, as it can be observed, the first one has the direction of maximum variance, Figure 1 (b) shows the projected data on this new basis, the projection is just a rotation of the axis. Figure 1 (c) shows the projection of the features over the first 2 eigenvectors, the X axis corresponds to the first feature and has the biggest variance.

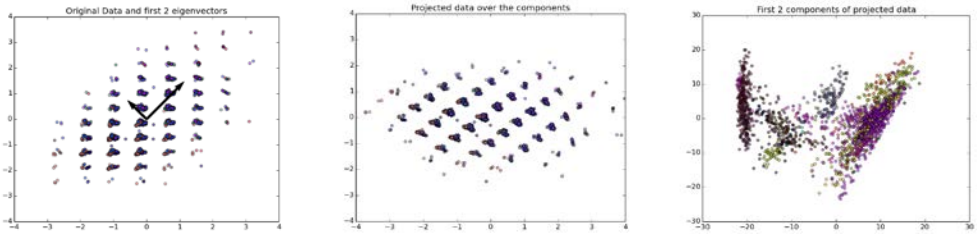

Figure 1: PCA projections.

Usually, not all the projected features are used, only those first components that explain most of the variance, leaving out those who barely have variance, the more components we use, the more variance explained and the lower the reconstruction error. Figure 2 (a) shows the ratio of variance explained with the number of components, it is observed that with only 50 features we can explain more than 99% of the variance. Figure 2 (b) shows the test accuracy for several learners when using different numbers of components, the graph shows that the more components are used, the better the accuracy. The best classifying algorithm is a crossvalidated SVM with Gaussian kernel as it was in [6], the accuracy of the system gets saturated at 50 components, same as the explained variance. It is interesting to notice that other algorithms such as LDA or LR does increase their performance after 50 features, even though they explain little variance and GNB reduces its accuracy, probably because the last features are correlated among them and independence assumption of GNB if not fulfilled.

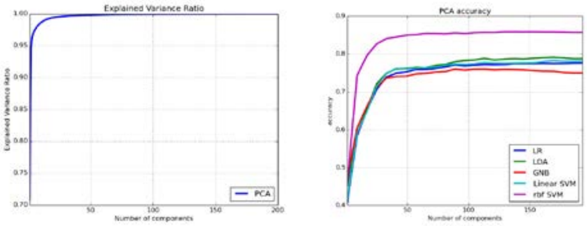

Figure 2: PCA explained variance and classification performance.

2.1 PLS

PLS is a supervised technique that finds the projections of the input and output data with maximum covariance $\bar{\bar{C}}_{XY}$, there are several implementations of this algorithm, Figure 3 shows the classification performance for 3 different implementations of PLS, SVD (a), Regression (b) and Cannonical (c).

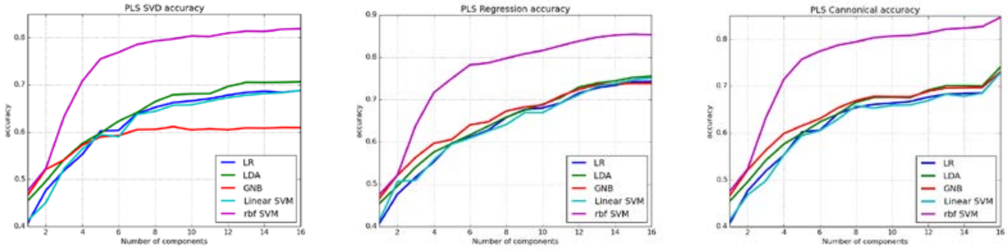

Figure 3: PLS implementation’s accuracy

As it can be observed, as we increase the number of extracted features, the accuracy increases. The SVM classifier with Gaussian kernel has by far the best accuracy; the other classifiers obtain similar score and behavior but score 10% less accuracy. The GNB classifier is particularly bad for the SVD implementation.

2.1 LDA

LDA is a supervised method that fits a Multivariate Gaussian density to each class, assuming that all classes share the same covariance matrix. Making that assumption, the decision boundaries are hyperplanes. This method finds the direction of minimum overlap among classes under these assumptions. LDA can be used as a linear feature extraction method, taking the decision boundary hyperplanes as the transforming matrix $\overline{\overline{U}}$.

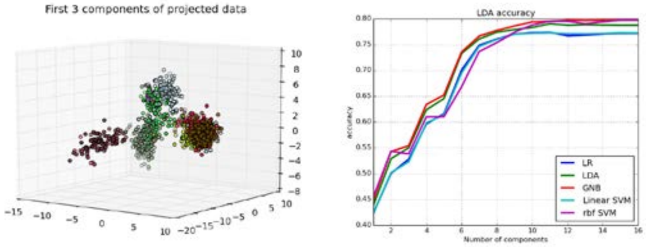

Figure 4: LDA data projection and accuracy

Figure 4 (a) shows the projected data over the first 3 components, we can observe the data is separated en cuasi-gaussian clusters of each class, meaning the projections form a nice separable feature space. Figure 4 (b) shows the accuracy of the different classifiers as we increase the number of features used. We can observe that these features have lost some relevant information since the best test accuracy has decreased 5% respect to the original data. LR and Linear SVM have exactly the same performance over the features and GNB reached maximum accuracy meaning the features are independent.

2.1 ICA

Given a set of linearly dependent variables, the goal of ICA is to find the transformation matrix $\overline{\overline{U}}$ that transforms them in a set of C linearly independent random variables. It is an unsupervised method in which the N D-divisional vectors form a basis of the subspace where the images are whitened and decorrelated.

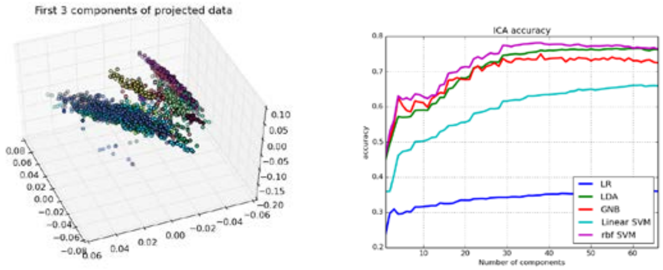

Figure 5: ICA data projection and accuracy

Figure 5 (b) shows the accuracy of the different classifiers as we increase the number of independent components the ICA algorithm must obtain, again LR and Linear SVM perform badly, LDA and SVM with Gaussian kernel obtain the maximum scores.

2.1 CCA

CCA is an algorithm that searches for the directions of maximum correlation $\overline{\overline{R}}_{XY}$ between input and output data. It usually outperforms PLS for classification, it generates as many features as classes. Figure 5 (b) shows the accuracy of the different classifiers as we increase the number of features used. A loss of relevant information is appreciated due to lower accuracy scores, LDA seems to work very well with this data and so does GNB, meaning that clusters are probably Gaussian and independent. SVM and LR perform surprisingly bad for this decomposition.

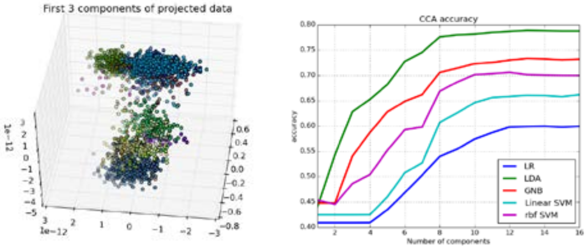

Figure 6: CCA data projection and accuracy

3 Kernel methods

Linear methods are simple, robust and lead to convex problems, but they lack expressive power. One common solution is to project the data into a higher dimensional space. The linear learning algorithm will operate with this non-linear transformation of the input feature space, thus performing non-linear classification over the initial features. Some kernels are i.e. the polynomic extension or the Gaussian kernel. The increase in dimension could cause overfitting and it increases the computational time and complexity of the classifier.

If the linear algorithm can be expressed in terms of inner products only, we can use the kernel trick by precomputing the distance in the transformed space between each training sample. The features of the training sample become the distance between the samples and the training samples in the transformed space. Gaussian kernels trick has been used for the PCA and PLS Regression feature extraction techniques, each sample vector now has as many features as training samples (2027), each feature is the distance between the initial sample and each training sample in the transformed feature space.

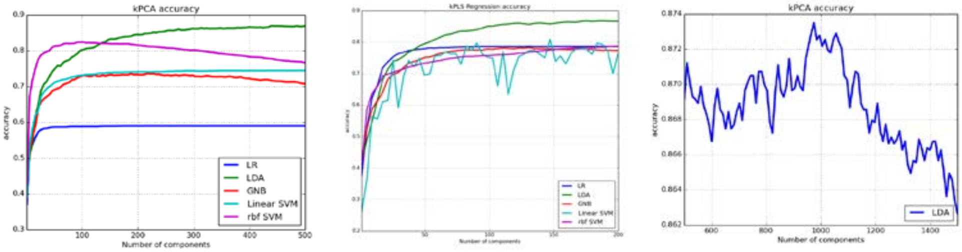

Figure 7: Kernel methods accuracy

As it can be observed in Figure 7, the kernel version of the algorithms affects each classifier's accuracy differently, in the case of SVM, it´s impoverished and sometimes it is unstable, probably because of a limited set of C values validated. LR and GNB has poorer performance due to the increased dimensionality of the problem and meaning that the features are nor independent**.** The LDA algorithm has the best performance, this means that the features generated cause each class to have multivariate Gaussian distribution with similar covariance matrix. For 950 features used, the LDA algorithm has an accuracy for testing of 87,3%, thus improving the 86% obtained in [6] and proving that feature extraction can improve the system's performance.

4 Conclusions

All the experiments performed point out the benefits of including a feature extraction module in a classification system. Linear methods PCA and PLS have proven to obtain the same accuracy as the initial dataset but with far less features, thus highlighting their dimensionality reduction capabilities. CCA and ICA didn't obtain as good results meaning that the performance of the techniques is highly dependent on the database. Kernel extensions combined with LDA makes an improvement of the accuracy in 1.7% with respect to [6], showing the capabilities and advantages of this technique, but it increases the computational cost of the system.

References

[1] C4.278.12996-1 MACHINE LEARNING APPLICATIONS 14/15-S2.

[2] AVIRIS dataset. http://aviris.jpl.nasa.gov/index.html

[3] Advanced Introduction to Machine Learning. 10715, Fall 2014. Eric Xing, Barnabas Poczos. School of Computer Science, Carnegie-Mellon University.

[4] scikit-learn library documentation http://scikit-learn.org/dev/index.html

[5] Chapter 9 DECISION TREES. Lior Rokach. Department of Industrial Engineering Tel-Aviv University

[6] Classifying systems on the AVIRIS dataset. Manuel Montoya Catalá. Master in Multimedia and Communications. Carlos III of Madrid University.