The Effect of Principal Component Analysis on Machine Learning Accuracy with High Dimensional Spectral Data

Tom Howley, Michael G. Madden, Marie-Louise O'Connell and
Alan G. Ryder, 2006

Abstract. The classification of high dimensional data, such as images, gene expression data and spectral data, poses an interesting challenge to machine learning, as the presence of high numbers of redundant or highly correlated attributes can seriously degrade classification accuracy. This paper investigates the use of Principal Component Analysis (PCA) to reduce high dimensional data and to improve the predictive performance of some well known machine learning methods. Experiments are carried out on a high dimensional spectral dataset, in which the task is to identify a target material within a mixture. These experiments employ the NIPALS (Non-Linear Iterative Partial Least Squares) PCA method, a method that has been used in the field of chemometrics for spectral classification, and is a more efficient alternative than the widely used eigenvector decomposition approach. The experiments show that the use of this PCA method can improve the performance of machine learning in the classification of high dimensional data

Presented by Aleksandr Tkachenko
March 26, 2008

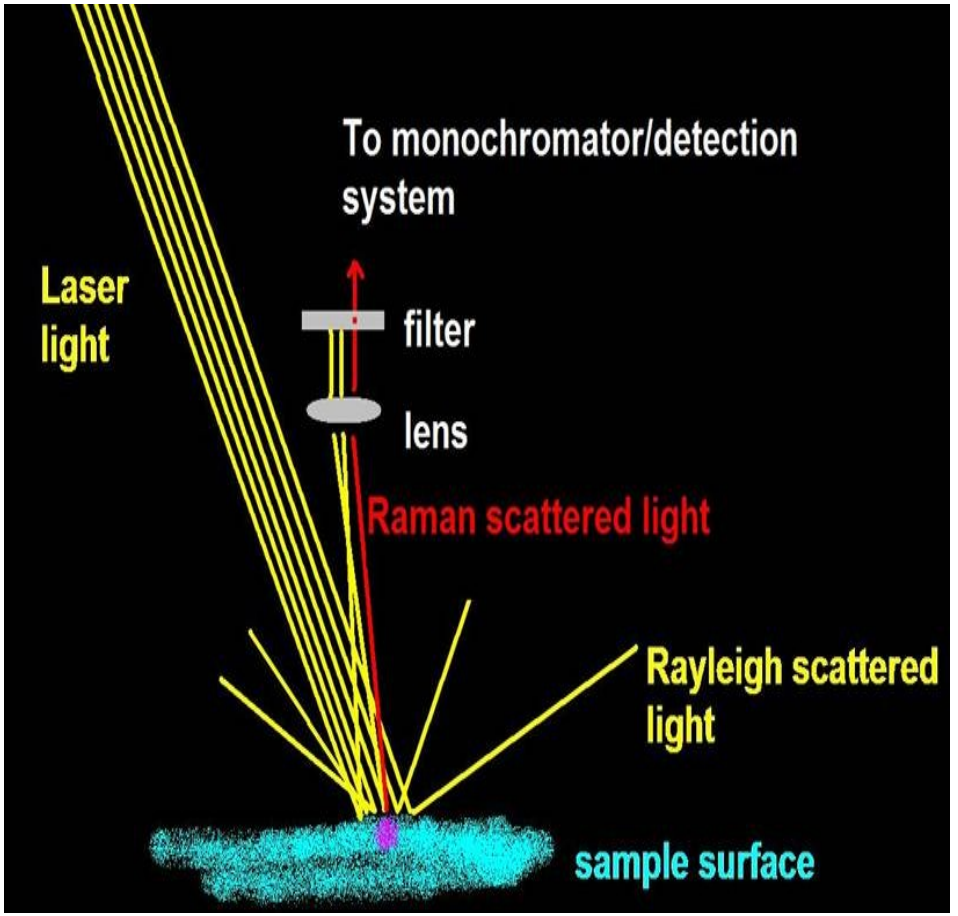
High-dimensional data

- Examples
  - Gene expression data
  - Images
  - Text documents
  - **Spectral data**
- Inherent properties
  - Redundancy
  - Correlation
  - Noise

Raman Spectroscopy

- Is used to study vibrational and rotational properties of molecules in materials

- Principle:
  - A monochromatic light source (laser) illuminates the sample
  - A small portion of the scattered radiation has frequencies different from that of the incident beam (Raman scattering)
  - This radiation contains information about the energies of molecular vibrations and rotations, which depend on the particular atoms or ions that comprise the molecule.



Raman Spectroscopy

- Applications:
  - Identification of narcotics and explosives
  - Detection of cancer in tissues
  - Monitoring of respiratory gas mixtures during surgery
  - Investigate the chemical composition of historical documents

Classification of Raman spectral data

- Representation of a data matrix:
  - Rows - samples/mixtures
  - Columns - all points on constituent spectra
  - Values – an intensity of scattering of the particular frequency in the particular sample
- The task is to identify mixtures containing a target material (e.g. cocaine)

Problematic aspects of Raman spectra

- *Collinearity:* many of the attributes (spectral data points) are highly correlated to each other which can lead to a degradation of the prediction accuracy.
- *Noise:* particularly prevalent in spectra of complex mixtures. Predictive models that are fitted to noise in a dataset will not perform well on other test datasets.
- *Fluorescence:* the presence of fluorescent materials in a sample can obscure the Raman signal and therefore make classification more difficult.
- *Variance of Intensity:* a wide variance in spectral intensity occurs between different sample measurements.
- *Overlapping Raman bands*
- *Residual laser scatter*

PCA

- Reduces the dimensionality of sample vectors
- Alleviates the problems of high-dimensionality and collinearity
- NIPALS
  - Non-Linear Iterative Partial Least Squares
  - Finds eigenvectors iteratively one by one
  - Effective for applications where the number PCs is known a priori

Dataset

- total samples (mixtures): 217
- 87 samples contain acetaminophen, a pain-revealing drug
- concentration of acetaminophen varies
- spectrum coverage: 350-2000 1/cm
- 1646 features per sample

Experiment setup

- Classifiers:
  - linear SVM
  - RBF SVM
  - k-Nearest Neighbour
  - decision tree
  - RIPPER (Repeated Incremental Pruning to Produce Error Reduction)
  - Naive Bayes
  - Linear Regression
- Weka implementation
- 10 fold cross validation
- Parameter tuning

Data preprocessing

- RD - raw dataset (no preprocessing)
- ND - dataset with each sample divided by the maximum intensity within the sample. Performing this step eliminated the large intensity difference between the spectra.
- FD - a Savitzky-Golay first derivative. Automated and reproducible method for background correction.
- FND - a normalization step is carried out after applying a first derivative to each sample of the raw dataset.

Results without PCA

Table 1. Percentage Classification Error of Different Machine Learning Methods on Acetaminophen Dataset

	Pre-processing Technique			
Method	**RD**	**ND**	**FD**	**FND**
Linear SVM	**6.45**	**2.76**	**3.23**	**0.92***
	(C=100)	*(C=1)*	*(C=10000)*	*(C=0.1)*
RBF SVM	**5.07**	**2.76**	**1.84**	**0.92***
	(C=1000, σ=0.1)	*(C=1000, σ=0.001)*	*(C=10000, σ=10)*	*(C=10, σ=0.01)*
k-NN	11.06	7.83	**4.61**	**4.15**
	(k=1)	*(k=1)*	*(k=10)*	*(k=1)*
C4.5	10.14	7.83	**1.84**	**1.38**
RIPPER	15.67	11.06	**3.69**	**2.3**
Naive Bayes	25.35	13.82	25.81	**5.53**
Linear Reg.	27.65	16.13	25.35	20.28

Results without PCA Conclusions

- SVMs
  - produce the best overall results
  - FD and FND doesn't improve performance significantly
  - Handle gracefully high degree of collinearity in the data
- Linear Regression
  - performs poorly with all preprocessing techniques
  - Fails to deal with correlated attributes
- Naive Bayes
  - high average error on the RD, ND and FD data
  - assumes independences of each of the attributes
- FD and FND improve the performance of the majority of the classifiers

Incorporating PCA

- Same classifiers and normalization techniques
- 10-fold cross-validation + PCA:

For each iteration of the 10-fold cross-validation repeat steps 1-5 :

1. Carry out PCA on the training data to generate a loadings matrix.
2. Transform training data into a set of PC scores using the first P components of the loadings matrix.
3. Build a classification model based on the training PC scores data.
4. Transform the held out test fold data to PC scores using the loadings matrix generated from the training data.
5. Test classification model on the transformed test fold.

Results with PCA

Table 2. Percentage Classification Error of Different Machine Learning Methods with PCA on Acetaminophen Dataset

	Pre-processing Technique			
Method	**RD**	**ND**	**FD**	**FND**
Linear SVM	5.07	**1.84**	**3.23**	**0.46**
	(P=18,C=0.1)	*(P=13,C=0.1)*	*(P=14,C=0.01)*	*(P=4,C=0.1)*
RBF SVM	6.91	**2.76**	**2.23**	**0.46**
	(P=19,C=100, σ=0.001)	*(P=16,C=10,* σ=0.001)	*(P=12, C=10,* σ=0.001)	*(P=5,C=10,* σ=0.001)
k-NN	11.06	5.99	**2.3**	**0.0***
	(P=17,k=3)	*(P=10,k=1)*	*(P=14,k=1)*	*(P=4,k=5)*
C4.5	7.83	7.37	7.37	**1.38**
	(P=20)	*(P=19)*	*(P=5)*	*(P=6)*
RIPPER	11.98	8.29	6.45	**2.3**
	(P=20)	*(P=8)*	*(P=5)*	*(P=3)*
Naive Bayes	38.71	10.6	11.52	3.23
	(P=1)	*(P=8)*	*(P=5)*	*(P=2)*
PCR	9.22	5.53	8.29	**1.38**
(PCA+Linear Reg.)	*(P=16)*	*(P=20)*	*(P=11)*	*(P=80)*

Results with PCA

- Conclusions:
  - FND improves the performance of the classifiers (except for Naive Bayes)
  - The same or numerically smaller error archived for all classifiers with FND preprocessing
  - Linear Regression gained the best improvement
  - SVN and k-NN demonstrate comparable performance with and without PCA

Effect of PCA on classification accuracy

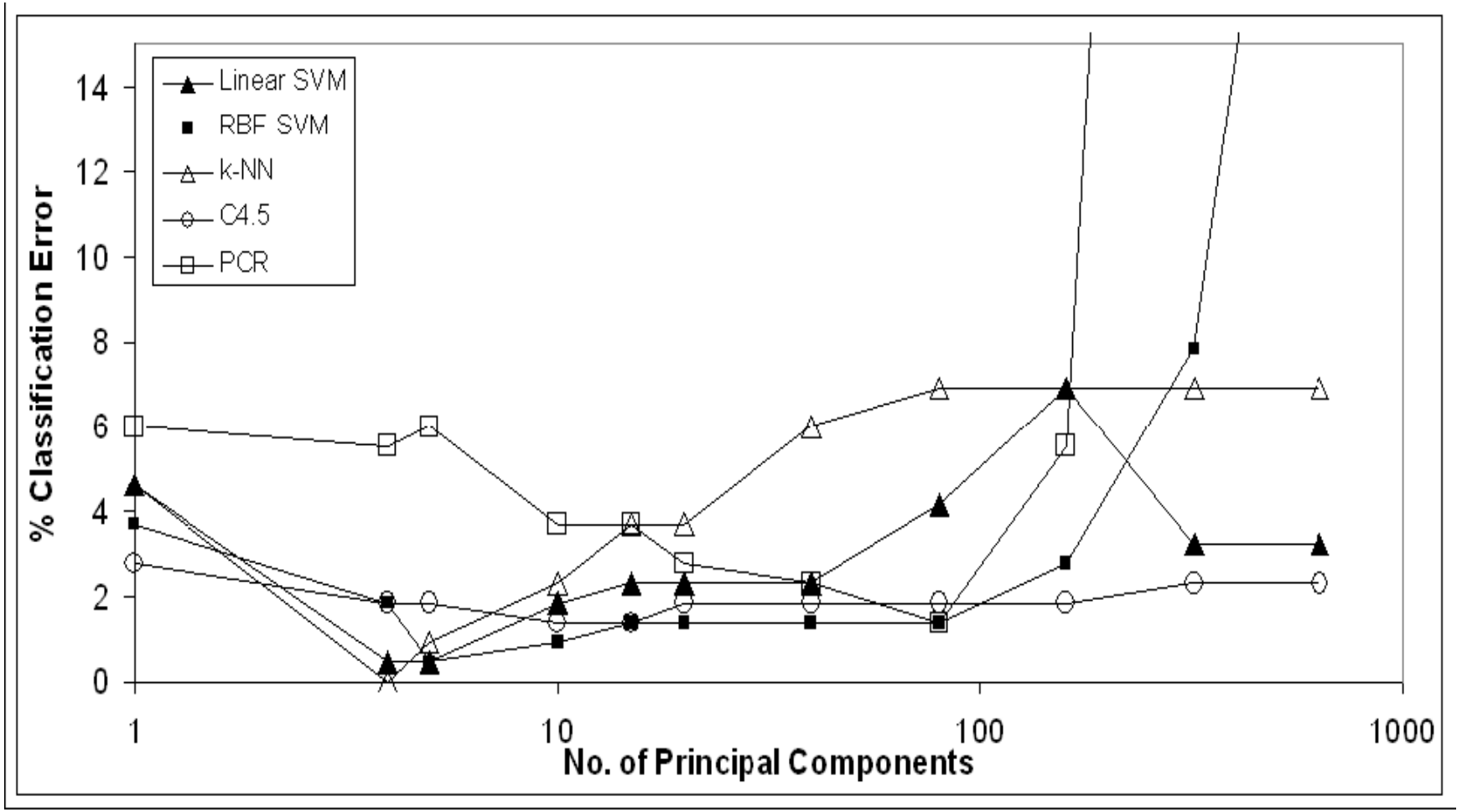


Fig. 1. Effect of changing the number of PCs on Machine Learning Classification Error. The best parameter setting and preprocessing technique was used for each classifier.

PCA vs original feature space

- Table 1 and Table 2 were compared using paired t-test with 5% confidence level
- Result: the overall improvement archived with PCA is significant

Conclusions

- PCA improves classification accuracy when applied to high-dimensional data.
- Only a small number of PCs capture the most of the variation (6/1646)!
- NIPALS is computationally feasible for PCA applications
- Preprocessing of first derivative followed by normalization improves the accuracy

Overall conclusion

NIPALS PCA + first derivative with normalization preprocessing appears to be a promising approach for the classification of high dimensional data.

Future work

- Test the approach on other high-dimensional datasets
- Investigate automatic selection of parameters