13

Dimensionality Reduction and Microarray data

David A. Elizondo, Benjamin N. Passow, Ralph Birkenhead, and Andreas Huemer

Centre for Computational Intelligence, School of Computing, Faculty of Computing Sciences and Engineering, De Montfort University, Leicester, UK,{elizondo,passow,rab,ahuemer}@dmu.ac.uk

Summary. Microarrays are being currently used for the expression levels of thousands of genes simultaneously. They present new analytical challenges because they have a very high input dimension and a very low sample size. It is highly complex to analyse multi-dimensional data with complex geometry and to identify low-dimensional "principal objects" that relate to the optimal projection while losing the least amount of information. Several methods have been proposed for dimensionality reduction of microarray data. Some of these methods include principal component analysis and principal manifolds. This article presents a comparison study of the performance of the linear principal component analysis and the non linear local tangent space alignment principal manifold methods on such a problem. Two microarray data sets will be used in this study. A classification model will be created using fully dimensional and dimensionality reduced data sets. To measure the amount of information lost with the two dimensionality reduction methods, the level of performance of each of the methods will be measured in terms of level of generalisation obtained by the classification models on previously unseen data sets. These results will be compared with the ones obtained using the fully dimensional data sets.

Key words: Microarray data; Principal Manifolds; Principal Component Analysis; Local Tangent Space Alignment; Linear Separability; Neural Networks.

13.1 Introduction

Microarray data is arrived at by using a high-throughput experimental technology in molecular biology. This data is used for parallel analysis of genes which may be involved in a particular disease. This high dimensional data is characterised by a very large variable/sample ratio. Typically, they contain a large number (up to tens of thousands) of genes, each expressed as a number. The number of samples, for each of these genes is relatively small (several tens). The high dimensionality of this data has two main consequences. On

the one hand, it makes its analysis challenging. On the other hand, intuitively, it might increase the likelihood that the data will be linearly separable.

The problem of high dimensionality can be approached with the use of dimensionality reduction methods. Principal component analysis and principal manifolds are commonly used methods for the analysis of high dimensional data. Principal manifolds were introduced by Hastie and Stueltze in 1989 as lines or surfaces passing through the middle of the data distribution [8]. This intuitive definition was supported by the mathematical notion of self-consistency: every point of the principal manifold is a conditional mean of all points that are projected into this point. In the case of datasets only one or zero data points are projected in a typical point of the principal manifold, thus, one has to introduce smoothers that become an essential part of the principal manifold construction algorithms.

One important application of principal manifolds is dimension reduction. In this field they compete with multidimensional scaling methods and the recently introduced advanced algorithms of dimension reduction, such as locally linear embedding (LLE) [12] and ISOMAP [19] algorithms. The difference between the two approaches is that the later ones seek new point coordinates directly and do not use any intermediate geometrical objects. This has several advantages, in particular that a) there is a unique solution to the problem (the methods are not iterative in their nature, there is no problem of grid initialisation) and b) there is no problem of choosing a good way to project points onto a non-linear manifold. This paper will use the principal manifold non-linear dimension reduction algorithm based on local tangent space alignment introduced in [23]. This method has been previously used to reduce the dimensionality of microarray data with good results [20]. The local tangent space alignment is a novel and interesting method which is available as a *ready to go* Matlab tool box [10] that has already been tested and verified.

Other work related to the analysis of Microarray data using dimensionality reduction techniques include [15], [11] and [21]. In [15] a semi-parametric approach is used to produce generalised linear models reducing the dimension, [11] uses graph theoretical methods to aid the search for models of reduced dimension and [21] uses discriminant partial least squares to provide models with more explanation of the response variables than might arise from the standard PCA method.

It is important to analyze the amount of information that is lost by the dimensionality reduction methods. This is why this article proposes the development of linear classifiers, using the fully dimensional and dimensionally reduced data sets, as a way to measure and compare the effects on the data caused by reducing the dimensionality. In the case of linearly separable data sets, several methods can be used to provide a separating hyperplane [3, 18]. When the sets are not linearly separable, a linear neural network such as the Recursive Deterministic Perceptron [5, 16, 17] or a Backpropagation Neural Network [13, 14] can be used.

In this study, two microarray data sets are used. The first data set is classified into three classification criteria. These include: five types of breast cancer type, positive or negative Estrogen-Receptor, and aggressive or non aggressive cancer. The second data set is classified into three types of bladder cancer.

This paper is divided into five sections. Some background information about dimension reduction and about the notion of linear separability are given in section two. This includes the introduction of a linear and a nonlinear dimensionality reduction methods, Principal Component Analysis and Local Tangent Space Alignment respectively. In section three, the procedure used to compare the two dimensionality reduction methods is presented with the use of two microarray data sets. Section four presents some results and discussion. A summary and some conclusions are presented in section five.

13.2 Background

In this section, some of the standard notions used throughout this chapter are introduced, together with some definitions and properties.

13.2.1 Microarray Data

In biological and medical research, DNA microarray technology is widely used to study gene expression in cells for example in the diagnosis of diseases including cancer. Therefore, this technology is a very important and widely used method in research and diagnosis. Unfortunately, the data produced by this method is highly dimensional. High dimensionality could mean tens or tens of thousands of dimensions, depending on the circumstances and experiment setup on which this data is produced. In this study, two microarray data sets, provided at the first workshop in principal manifolds[1] held in Leicester in 2006, were used. The first data set [22], here after referred to as D1, was initially used in breast cancer research to identify patterns of breast cancer gene expressions that could be used to predict the patients disease progression. The data set consists of 17816 gene expressions. As a standard procedure, the data is preprocessed in such a way that the absolute average expression level is zero. This is done because only the difference in expression between samples, as opposed to the overall gene expression, contains useful biological meaning . This data set contains a total of 286 samples which correspond to the number of patients from which the samples were taken.

The second data set used in this study [6], here after referred to as data set D2, was originally used in bladder cancer research. The data set consists of 3036 gene expressions, also preprocessed to have zero mean as in the case of the D1 data set. The number of samples in this data set is 40.

[1] http://www.ihes.fr/∼zinovyev/princmanif2006/

One of the features of the data sets is that their dimensionality is much higher than the sample size. This makes the analysis extremely difficult if no dimensionality reduction method is applied beforehand. Therefore, this kind of data is mostly reduced into a denser representation, keeping only the "most important" aspects of the data. The number of available methods that can be used for this reduction is growing, especially because there is no "correct" solution possible due to the fact that some information is always lost in the process.

13.2.2 Methods for Dimension Reduction

Several methods for the analysis of high dimensional data have been proposed including principal components analysis (linear) and principal manifolds (non linear). In this study, the level of performance of the principal component analysis (PCA) and the local tangent space alignment (LTSA) non linear principal manifold methods is studied for dimensionality reduction of high dimensional microarray data.

Principal Component Analysis

The PCA dimensionality reduction method [9, 7] is a linear dimensionality reduction method. It works by projecting a number of correlated variables into a (smaller) number of uncorrelated variables called principal components. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible. The algorithm solves for the eigenvalues and eigenvectors of a square symmetric matrix with sums of squares and cross products. The eigenvector associated with the largest eigenvalue has the same direction as the first principal component. The eigenvector associated with the second largest eigenvalue determines the direction of the second principal component. The sum of the eigenvalues equals the trace of the square matrix and the maximum number of eigenvectors equals the number of rows (or columns) of this matrix.

Local Tangent Space Alignment

The Local Tangent Space Alignment algorithm was introduced in [23]. The LTSA is a nonlinear dimensionality reduction method that aims to find a global coordinate system within a low dimensional space that best characterises the high dimensional data set. It finds an approximation to the tangent space at each point using a neighbourhood and then aligns these tangent spaces in a process of constructing the coordinate system for the non-linear manifold. The computation speed is affected by the choice of size of neighbourhood due to the search cost for the nearest neighbours. There was an interaction between the LTSA output and the perceptron algorithm used for finding

the separating plane. The perceptron algorithm took longer to converge in cases where the LTSA was applied with small neighbourhoods, requiring a trade off to get improved performance.

Strengths and Limitations of the PCA and LTSA Methods

Currently there is no general method available to distinguish the nature of data as being linear or non linear [20]. The high dimensional space and low number of samples found in microarray data sets makes it appear highly likely to be linearly separable data and hides away any non linear structures. This is why the study of both linear and non linear dimensionality reduction methods is interesting.

PCA is a well established method that has been used over many years and frequently on Microarray data. According to [20] the PCA is fast to compute and easy to implement. Its complexity of the PCA algorithm is $O(n_s \times n)$[2] where n_s represents the sample size, and n the original dimensionality. This method is guaranteed to find a lower dimensional representation of the data on a linear subspace if such representation exists. However, as mention in [1] the PCA method can only identify gross variablity as opposed to distinguishing among and within groups variability.

In [20], the LTSA method has been used to reduce the dimensionality of microarray data with good results and it outperformed the linear PCA method in some aspects. The LTSA is a fairly recent method that reduces the high dimensionality on data sets by using tangent space produced by fitting an affine subspace in the proximity of each data sample. The LTSA algorithm is using a k-nearest neighbours search that can be computational expensive for large k and large input matrices. A small neighbourhood can result in less accurate dimensionality reduction. Also, the computation of the smallest eigenvectors of the alignment matrix used by the algorithm, is computationally expensive. Therefore, the PCA is more computationally efficient than the LTSA. Nevertheless, the non linear structures intrinsic in the data can not be efficiently exploited by using the linear PCA method.

13.2.3 Linear Separability

Preliminaries

The following standard notions are used: Let $\boldsymbol{p}_1, \boldsymbol{p}_2$ be the standard position vectors representing two points P_1 and P_2 in $\mathbb{R}^d$,

- The set $\{t\boldsymbol{p}_1 + (1-t)\boldsymbol{p}_2 \mid 0 \leq t \leq 1\}$ is called the segment between $\boldsymbol{p}_1, \boldsymbol{p}_2$ and is denoted by $[\boldsymbol{p}_1, \boldsymbol{p}_2]$.
- The dot product of two vectors $\boldsymbol{u} = (u_1, ..., u_d), \boldsymbol{v} = (v_1, ..., v_d)$ is defined as $\boldsymbol{u}^T\boldsymbol{v} = u_1v_1 + ... + u_dv_d$. $Adj(\boldsymbol{u}, r) = (u_1, ..., u_d, r)$ and by extension $Adj(S, r) = \{Adj(\boldsymbol{x}, r) \mid \boldsymbol{x} \in S\}$.

- $\mathcal{P}(\boldsymbol{w}, t)$ stands for the hyperplane $\{\boldsymbol{x} \in I\!R^d \mid \boldsymbol{w}^T\boldsymbol{x} + t = 0\}$ of $I\!R^d$. $\boldsymbol{w}$ is the normal (i.e. is perpendicular), to the hyperplane $I\!P$. The threshold t is proportional to the distance from the origin to $I\!P$. $I\!P$ will stand for the set of all hyperplanes of $I\!R^d$.

Two sub-sets X and Y of $I\!R^d$ are said to be linearly separable (LS) if there exists a hyperplane P of $I\!R^d$ such that the elements of X and those of Y lie on opposite sides of it. Figure (13.1) shows an example of both a LS (a) and a NLS (b) set of points. Squares and circles denote the two classes.

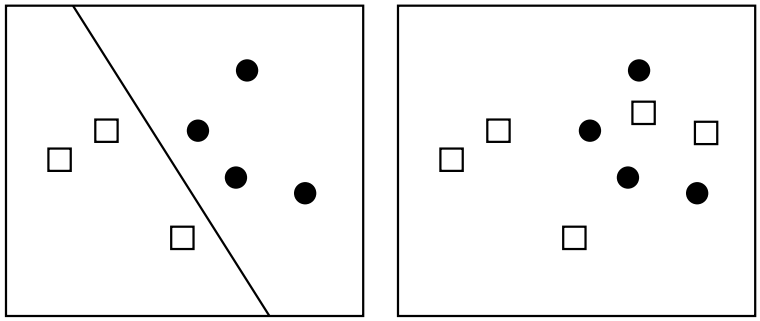

Fig. 13.1. LS (**a**) and a non-LS (**b**) set of points

Methods for testing linear separability

The methods for testing linear separability between two classes can be divided into five groups:

- **The methods based on solving systems of linear equations.** These methods include: the Fourier-Kuhn elimination algorithm, and the Simplex algorithm. The original classification problem is represented as a set of constrained linear equations. If the two classes are LS, the two algorithms provide a solution to these equations.
- **The methods based on computational geometry techniques.** The principal methods include the convex hull algorithm and the class of linear separability method. If two classes are LS, the intersection of the convex hulls of the set of points that represent the two classes is empty. The class of linear separability method consists in characterising the set of points P of $I\!R^d$ by which it passes a hyperplane that linearly separates two sets of points X and Y.

- **The methods based on neural networks.** The perceptron neural network is one of most commonly used methods for testing linear separability. If the two classes are LS, the perceptron algorithm is guaranteed to converge, after a finite number of steps, and will find a hyperplane that separates them.
- **The methods based on quadratic programming.** These methods can find a hyperplane that linearly separates two classes by solving a quadratic optimisation problem. This is the case for the Support Vector Machines.
- **The Fisher Linear Discriminant method.** This method tries to find a linear combination of input variables, $w \times x$, which maximises the average separation of the projections of the points belonging to the two classes C_1 and C_2 while minimising the within class variance of the projections of those points.

These methods are described in detail in [4]. Several heuristic methods, to reduce the calculation time while testing for linear separability, are presented in [3]. Table 13.1 describes the perceptron algorithm which is one of the most commonly used algorithms for testing linear separability. A faster more efficient algorithm for testing linear separability on relatively small data sets is the Simplex. However, due to the high dimensionality of the microarray data sets, the use of the Simplex algorithm becomes impractical. This is why the selected method for testing linear separability, used for this work, is the Perceptron algorithm.

Table 13.1. The Perceptron Learning algorithm.

PERCEPTRON$(S, \boldsymbol{w})$
– data: a set of vectors S constructed from the classes X and Y we wish to distinguish
– result: a weight vector $\boldsymbol{w}$ which separates the two classes if they are LS
$\boldsymbol{w}_0 := \boldsymbol{x}_0; (\boldsymbol{x}_0 \in I\!R^{d+1})$
$k \quad := 0;$
while $(\exists \boldsymbol{x}_k \in S)$ such that $(\boldsymbol{w}_k^T \boldsymbol{x}_k \leq 0)$ **do**
Begin
$\boldsymbol{w}_{k+1} := \boldsymbol{w}_k + \boldsymbol{x}_k;$
$k \quad := k + 1;$
End

The problem of finding a hyperplane $\mathcal{P}(\boldsymbol{w}, t)$ which separates sets X and Y in $I\!R^d$ [i.e. finding $\boldsymbol{w}, t$ where $\boldsymbol{w}^T \boldsymbol{x} + t > 0$ and $\boldsymbol{w}^T \boldsymbol{y} + t < 0$ for all

$\boldsymbol{x} \in X, \boldsymbol{y} \in Y]$ is equivalent to finding $\boldsymbol{w}_1 \in I\!R^{d+1}$ for which $(\boldsymbol{w}_1^T \boldsymbol{s} > 0 \ \ \forall \boldsymbol{s} \in S$ where $S = Adj(X, -1) \cup -Adj(Y, -1)$. [Given $\boldsymbol{w}_1$ which solves the "S problem" in $I\!R^{d+1}$, the separability hyperplane in $I\!R^d$ is $\mathcal{P}(\boldsymbol{w}, t)$ where $\boldsymbol{w}_1 = Adj(\boldsymbol{w}, -t)$. It can be seen that $\boldsymbol{w}^T \boldsymbol{x} + t > 0$ and $-\boldsymbol{w}^T \boldsymbol{y} - t > 0 \Rightarrow \boldsymbol{w}^T \boldsymbol{y} + t < 0$]. The presented perceptron algorithm is implemented in $I\!R^{d+1}$ solving the problem of S. The solution to the original problem for X and Y is then obtained.

13.3 Comparison Procedure

The comparison performed in this study was realised using microarray data sets together with their given classifications. These sets were split up into training and testing data sets, 60% for training and 40% for testing. For the classification model a neural network was first developed with the training set and then tested with the testing set. This procedure was also done with two dimensionality reduction methods before the data set was split and the classification model was used. The dimensions of the data sets were reduced by using principal component analysis (PCA) and the local tangent space alignment (LTSA) manifolds introduced in section two. The level of performance of the dimensionality reduction methods was measured in terms of the level of generalisation obtained and the time to compute needed by the classification models on previously unseen data sets. The system used for this comparison was a modern standard PC with Matlab installed. All computation and implementation of algorithms was realised in Matlab using standard toolboxes as well as the toolbox_dimreduc - a toolbox for dimension reduction methods [10] . The following section will explain the complete comparison procedure in more detail.

13.3.1 Data Sets

Microarray data set D1 was initially used to identify patterns of breast cancer gene expressions that could be used to predict the patients disease progression. It is a collection of 17816 gene expressions with a sample size of only 286. Three ab initio sample classifications are available with the data set:

- **Group** non-aggressive (A) vs aggressive (B) breast cancer
- **ER** estrogen-receptor positive (ER+) vs negative (ER-) tumours
- **Type** lumA, lumB, normal, errb2, basal and unclassified breast cancer types

The table 13.2 gives an overview of the classes, the number of members of each class as well as the number of members of other classes. As a regular perceptron was used in this work to classify the sets, only one class versus the rest of the classes could be handled at one time. Therefore the type classification was done by training and testing each of the six classes separately.

The other data set used for the comparison task is the data set D2, identified in section 2. The data set has 3036 gene expressions with a sample size of only 40. It was originally analyzed to gather knowledge on bladder cancer. The available classification used in this work is the clinical categorisation of the tumour into three classes.

- **Type** T1, T2+ and Ta tumour types

Table 13.2. DNA Microarray Data Set D1 Classes

Data Set	Classification	Classes	Samples per class	Rest
D1	Group	non-aggressive (A)	193	93
		aggressive (B)	93	193
	Estrogen-Receptor	positive (ER+)	209	77
		negative (ER-)	77	209
	Cancer type	lumA	95	191
		lumB	25	261
		normal	35	251
		errb2	34	252
		basal	55	231
		unclassified	42	244
D2	Tumour type	T1	11	29
		T2+	9	31
		Ta	20	20

Cross Validation

In this study the technique of cross validation was applied to split the data into training and testing data sets. In detail, ten equally sized classification data sets were made, sixty percent of the samples were used for training the neural networks and the remaining forty percent were used for testing purposes. The ten sets were made out of the original data set by moving a window 10 percent for each new set. The resulting ten sets each include all data from the original set but with the training and testing sets varying. Thus, for each of the two dimensionality reduction methods, ten different neural networks were developed and tested using different combinations of test sets that were picked up from the ten similar sample sets. Without this method the single test result could be interpreted in a wrong way due to e.g. an uneven distribution of samples in the set. If the method of cross validation is applied, one can use statistical methods of the multiple results to provide a more general interpretation of the results.

Sample Distribution

The microarray data set D2 was originally sorted with respect to their membership to the classes. Simply splitting up the data set into training and testing sets would result in sets missing a complete class or containing samples of only a single class. In order to solve this problem the data was randomised into sets with the constraint that at least three members of any class were present in each train and test data set.

13.3.2 Dimensionality Reduction

In general, high dimensional data, especially combined with a low sample size, can make further analysis with the raw data computational expensive and difficult. For comparison purposes in this work, two dimensionality methods were chosen, the principal component analysis and the local tangent space alignment. These two methods represent a well known linear approach and a nonlinear manifold learning approach.

Principal Component Analysis (PCA)

PCA is a method well known and frequently used for dimensionality reduction. It projects the high dimensional data onto a new coordinate system with fewer dimensions. The highest amount of the variance of the original data is taken care of by the first coordinate or principal component, the second highest amount by the second principal component and so on. The matlab function princomp was used for this study. Due to the very high dimensionality, the flag *econ* had to be used when executing the princomp function. This flag restricts the computation and output to only include the eigenvalues of the covariance matrix of the input data that are not necessarily zero. Without this flag set, the returned principal component coefficients matrix itself would have taken 2.4 GB of RAM, exceeding the memory available to Matlab. In order to compare the dimensionality reduction methods the execution time for the princomp function was measured and noted. After the dimensionality reduction took place, the chosen number of dimensions to be used for further computation correspond to 80 percent of the total variance of the components, the eigenvalues of the covariance matrix. For the data set D1, 80 percent correspond to 150 dimensions. For the data set D2, 80 percent correspond to 19 dimensions. The preprocessed data returned by *princomp* was then split into the training and test sets that were subsequently used to train and test the perceptron.

Local Tangent Space Alignment (LTSA)

The LTSA technique used in this work is a nonlinear manifold learning method. The algorithm is part of the toolbox_dimreduc available for Matlab.

Using this implementation, two parameters need to be specified next to the input matrix: the dimensionality of the output matrix as well as the number of nearest neighbours to take into account. The correct choice of parameters is important to achieve a good result within a reasonable time frame.

A careful choice of the number of output dimensions is important too. The parameters used for the data sets and classifications are listed in table 13.3. Due to the huge size of the D1 data set, the parameters for it were chosen by hand using trial and error. The parameters for the D2 data set on the other hand were searched for by a script checking a large number of sensible parameter combinations. In the end the best solution was chosen. As part of the comparison process, the time needed for the LTSA to execute was measured and noted. Thereafter the resulting data was split up into training and testing data sets for the perceptron.

Table 13.3. LTSA parameters used

Data set	Classification	Dimensions	Neighbours
D1	Group	114	133
	Estrogen - Receptor	80	99
	Cancer type	90	131
D2	Tumor type	18	31

13.3.3 Perceptron Models

The classification model is using a simple feedforward artificial neural network, a perceptron. It is capable of solving any linear separable classification problem in a limited time frame. The perceptron takes the two classes from the training data set and develops the network weights and bias accordingly. The iteration limit for the implemented perceptron was set to 10,000. The time the network needed to calculate the weights was measured and noted. Next, the network was tested using the testing data set. The level of generalisation, the percentage of successful classifications by the sum of all classifications, was computed and noted as well.

This procedure was used to compare the generalisation and the time needed to compute the data of the original high dimensional microarray data and the data sets reduced in dimensionality with the PCA and the LTSA methods beforehand. This process is repeated for all of the ten cross validation data sets, for all classes of all classifications, and the two data sets.

13.4 Results

This section presents results on the comparison of the convergence time and the level of generalisation obtained with the classification models created using

raw and dimensionality reduced microarray data sets. Since both in high and low dimensions the data sets are linearly separable, a single layer Perceptron neural network was used to create the classification models. The technique of cross validation was applied to split the microarray data sets into training and testing data sets. Table 13.4 shows the times needed for each manifold method to reduce the dimensionality of the data sets. As seen before, the PCA method produces more dimensions than the LTSA. However, the convergence time of the PCA method is less than 10% of that of the LTSA one. The PCA convergence times are similar for each data set. The LTSA method shows convergence time differences due to the choice of different dimensions and nearest neighbours taken into account (see table 13.3).

Table 13.4. Dimensionality reduction time taken by each manifold method (seconds)

Data Set	Class	PCA	LTSA
D1	Type	9.96	165.78
	Group	10.16	169.80
	ER+/-	9.86	138.90
D2	Type	0.109	0.697

Table 13.5 shows the average convergence time (across ten cross validation data sets), required to train the Perceptron neural networks using raw and PCA and LTSA preprocessed microarray data sets. It can be clearly seen that both PCA and LTSA give a dramatic improvement in the construction of the Perceptron classification neural networks for the D1 data set with mean differences ranging from 4.8 up to 45 times faster. This is expected as the original number of dimensions was dramatically cut down using the two dimensionality reduction methods. Overall, the average smaller convergence times were obtained with the PCA preprocessed data sets. For the D2 data sets, this difference is not as conclusive with convergence times ranging from 0.15 to 2 times faster. The D2 data set has only 24 training samples compared to 172 for the D1 data set. This could lead, depending on the distribution of the samples in the space, to more cycles of the perceptron algorithm for the D1 data set compared to the D2 one.

Although the times of convergence obtained with the perceptron neural network using the PCA and the LTSA dimensionality reduced data sets are not as clear in difference, the time to reduce the dimensionality using these two methods is large, with the PCA being, on average, 15 times faster for the D1 data set, and 7 times faster for the D2 data set as seen in table 13.4. The LTSA reduces the dimensions of the original data sets to less than the PCA method. This is due to the choice of dimensions and nearest neighbour parameters for the LTSA method as introduced previously in table 13.3. Better results with the perceptron neural network were obtained by fine tuning these parameters.

For the data set D1 it can be seen that it is more time efficient to reduce the dimensionality with a PCA beforehand training the perceptron. The overall time to reduce the dimensions with the PCA and train the perceptron is less than training the perceptron on the original data set. This is due to the many fewer dimensions the perceptron needs to take into account when finding the hyperplane which linearly separates the different classes. For the data set D1, the dimensions left, after applying the PCA, are less than one percent of the original data set.

The LTSA dimensionality reduction is more time consuming than the PCA method. However, when classifying multiple classes of the D1 data set, applying the LTSA dimension reduction method, prior to training the perceptron, is more time efficient as well.

Because of the smaller dimensionality of the D2 data set, the results, in terms of time, are not as conclusive as the ones obtained with the D1 data set.

Table 13.5. Results obtained in terms of time needed to train (convergence time) the Perceptron neural network. All values in seconds

	No Dim. Reduction				**PCA**				**LTSA**			
Data set	Min	Mean	Max	Mode	Min	Mean	Max	Mode	Min	Mean	Max	Mode
D1 - Type:												
Class 1 vs Rest	12.43	18.15	29.81	12.43	0.37	0.46	0.55	0.37	2.69	3.78	4.44	2.69
Class 2 vs Rest	71.83	100.15	148.82	71.83	1.23	2.30	2.86	1.23	0.53	0.97	1.69	0.53
Class 3 vs Rest	62.69	85.03	122.01	62.69	1.29	2.66	4.32	1.29	1.21	3.04	4.05	1.21
Class 4 vs Rest	77.87	86.19	97.47	77.87	1.26	1.89	2.77	1.26	0.82	1.38	2.28	0.82
Class 5 vs Rest	10.94	16.14	24.70	10.94	0.35	0.43	0.53	0.35	0.65	0.93	1.30	0.65
D1 - ER+/-:												
Class 1 vs Class 2	23.07	32.48	45.26	23.07	0.68	0.89	1.14	0.68	3.29	16.28	32.71	3.29
D1 - Group:												
Class 1 vs Class 2	58.66	69.14	87.39	58.66	1.38	1.97	2.91	1.38	3.10	10.19	17.90	3.10
D2 - Type:												
Class 1 vs Rest	0.035	0.055	0.065	0.035	0.019	0.028	0.033	0.019	0.016	0.028	0.043	0.016
Class 2 vs Rest	0.054	0.070	0.084	0.054	0.027	0.067	0.178	0.027	0.020	0.072	0.272	0.020
Class 3 vs Rest	0.056	0.068	0.081	0.056	0.066	0.170	0.492	0.066	0.222	0.406	0.651	0.222

Table 13.6 shows the level of generalisation, in terms of percentage of well classified samples, obtained using the raw, and PCA/LTSA preprocessed data sets. The generalisation results are presented in terms of the mean, mode (the most frequently occurring value), min, and max obtained over the ten data subsets generated with the cross validation approach.

This table shows that the classification results obtained with the dimensionality reduction methods are generally better than the ones obtained with the raw data sets. The results obtained with the PCA preprocessed data sets are significantly higher than the ones obtained with the LTSA reduction method. However, in terms of the number of dimensions, the LTSA performance is better than the one obtained with the PCA method. For example, in the data set D1, the variance of the PCA transformed data accomplishes for 80%. In the case of the D1 cancer type classification, 90 dimensions corre-

spond to 67% of the total variance with regards to the PCA method. Overall slightly better results are obtained with the LTSA with respect to the raw data sets.

The best generalisation results are obtained with classes 1 and 5 of the D1 data set. These two classes have the highest number of samples compared to the other classes (see table 13.2). Therefore, more data was available to train the perceptron linear classification models. The level of generalisation obtained with the LTSA preprocessed data on class 5 is surprisingly low in comparison with both raw and PCA preprocessed data sets. This is the only class where such large differences, in terms of the level of generalisation, can be seen. This might suggest that the LTSA parameters have to be fine tuned independently for each of the classes.

All methods obtained the same maximum level of generalisation for class 1 of the D2 data set. LTSA gives higher results for class 2. The raw data provides the best maximum generalisation level for class 3. Overall, the PCA provides better results closely followed by those obtained with the LTSA method.

Table 13.6. Results obtained with the Perceptron in terms of the level of generalisation

	No Dim. Reduction				**PCA**				**LTSA**			
Class	Min	Mean	Max	Mode	Min	Mean	Max	Mode	Min	Mean	Max	Mode
D1 - Type:												
Class 1 vs Rest	70.18	74.30	79.82	75.44	72.81	76.75	79.82	75.44	59.65	74.47	84.21	77.19
Class 2 vs Rest	7.89	16.93	24.56	18.42	26.32	34.04	37.72	36.84	9.65	16.40	28.95	9.65
Class 3 vs Rest	12.28	20.26	28.95	20.18	28.95	33.86	39.47	33.33	20.18	23.86	29.82	21.93
Class 4 vs Rest	15.79	21.93	28.07	19.30	32.46	38.07	43.86	34.21	16.67	23.33	30.70	24.56
Class 5 vs Rest	75.44	83.60	89.47	86.84	77.19	83.33	89.47	80.70	19.30	22.81	26.32	21.93
D1 - ER+/-:												
Class 1 vs Class 2	58.77	65.00	73.68	66.67	59.65	68.25	79.82	71.05	43.86	52.72	60.53	43.86
D1 - Group:												
Class 1 vs Class 2	42.11	48.86	56.14	43.86	49.12	54.91	60.53	53.51	45.61	53.77	63.16	54.39
D2 - Type:												
Class 1 vs Rest	25.00	54.38	81.25	56.25	50.00	65.63	81.25	62.50	31.25	57.50	81.25	37.50
Class 2 vs Rest	18.75	32.50	56.25	25.00	31.25	44.38	56.25	37.50	18.75	40.00	62.50	37.50
Class 3 vs Rest	31.25	46.88	68.75	43.75	25.00	40.00	56.25	43.75	50.00	54.38	62.50	50.00

13.5 Conclusions

A comparison study of the performance of the linear principal component analysis and the non linear local tangent space alignment principal manifold methods was presented. Two microarray data sets were used in this study. The first data set contained five, two and two classes. The second data set contained three classes. Linear classification models were created using fully dimensional and dimensionality reduced data sets. To measure the amount of information lost with the two dimensionality reduction methods, the level of performance of each of the methods was measured in terms of level of

generalisation obtained by the classification models on previously unseen data sets.

In terms of convergence time, the benefit offered by the dimensionality reduction methods is clear. Using the PCA dimensionality reduction method, prior to the development of the classification models, is over all faster than developing the models with raw, fully dimensional, data. For the LTSA, the time benefit is less significant. Nevertheless, for training multiple classification models, the LTSA is also a time beneficial alternative. This was shown to be the case for both microarray data sets.

In terms of generalisation, the linear classification models, built using both PCA and LTSA dimensionality reduction methods, frequently outperform the ones developed using raw data. The models developed by using the PCA method, give more consistent results than the ones using the LTSA. However, for this microarray data sets, the LTSA method produces more compact reduced data sets.

In conclusion, the results obtained with this study, do not allow to clearly measure the amount of information lost by the dimensionality reduction methods. Nevertheless, the results obtained are interesting, demonstrating conclusively that the level of generalisation was better when using dimensionality reduction methods. The reason for this might be related to the level of noise in the data. Most of the variance in the original 17816 dimensional data is provided by only about 150 dimensions. Due to the the high number of irrelevant dimensions, the inherited linear separability of these data sets might hide away non linear structures in the data.

Further studies could include the use of other manifold methods (linear and non linear). The use of non linear methods for building the classification methods might also be of interest since the current linear separability of the original data might be related to the few samples in a large input space. Data sets containing more samples will probably result in better classification models.

References

1. Barker, M. and Rayens, W.: Partial least squares for discrimination. Journal of Chemometrics, **17**, 166–173 (2002)
2. Bollacker, K.D. and Ghosh, J.: Linear feature extractors based on mutual information. In: Proceedings of the 13th International Conference on Pattern Recognition, volume 2, pages 720–724, Vienna, Austria, (1996)
3. Elizondo, D.: Searching for linearly separable subsets using the class of linear separability method. In: IEEE-IJCNN'04, pages 955–960 (2004)
4. Elizondo, D., The linear separability problem: Some testing methods. Accepted for Publication: IEEE TNN (2006)
5. Elizondo, D., Birkenhead, R., and Taillard, E.: Generalisation and the recursive deterministic perceptron. In: IEEE IJCNN'06, (2006)

6. Dyrskjot, L., Thykjaer, T., Kruhoffer, M. et al.: Identifying distinct classes of bladder carcinoma using microarrays. Nat Genetics **33** (1), 90–96 (2003).
7. Fukunaga, K.: Introduction to Statistical Pattern Recognition. Elsevier (1990)
8. Hastie, T. and Stuetzle, W.: Principal curves. Journal of the American Statistical Association, **84** (1989)
9. Pearson, K.: On lines and planes of closest fit to systems of points in space. Philosophical Magazine, **2** (6), 559–572 (1901)
10. Peyré, G.: http://www.mathworks.com/matlabcentral/fileexchange/. Dimension reduction toolbox (2006)
11. Castelo, R. and Roverato, A.: A robust procedure for gaussian graphical model search from microarray data with p larger than n. Journal of Machine Learning Research, **7**, 2621–2650 (2006)
12. Rowels, S. and Saul, L.: Non linear dimensionality reduction by locally linear embedding. Science, **290** (2000)
13. Rumelhart, D. E., McClelland, J. L., and the PDP Research Group: Parallel Distributed Processing, vol. 1. The MIT Press, Cambridge, MA (1986)
14. Rumelhart, D. E., McClelland, J. L., and the PDP Research Group: Parallel Distributed Processing, volume 2. The MIT Press, Cambridge, Massachusetts (1986)
15. Lambert-Lacroix, S. and Peyre, J.: Local likelihood regression in generalized linear single-index models with applications to microarray data. Computational Statistics and Data Analysis, **51** (3), 2091–2113 (2006)
16. Tajine, M. and Elizondo, D.: The recursive deterministic perceptron neural network. Neural Networks, **11**, 1571–1588 (1997)
17. Tajine, M. and Elizondo, D.: Growing methods for constructing recursive deterministic perceptron neural networks and knowledge extraction. Artificial Intelligence, **102**, 295–322 (1998)
18. Tajine, M. and Elizondo, D.: New methods for testing linear separability. Neurocomputing, **47**, 161–188 (2002)
19. Tenenbaum, J. B., de Silva, V., and Langford, J. C.: A global geometric framework for nonlinear dimensionality reduction. Science, **290** (5500), 2319–2323, (2000)
20. Teng, L., Li, H., Fu, X., Chen, W., and Shen, I.: Dimension reduction of microarray data based on local tangent space alignment. In: Fourth IEEE Conference on Cognitive Informatics, pages 154– 159. University of California, Irvine, USA, August (2005)
21. Tan, Y., Shi, L., Tong, W., Hwang, G.T.G., and Wang C.: Multi-class tumor classification by discriminant partial least squares using microarray gene expression data and assessment of classification models. Computational Biology and Chemistry, **28** (3), 235–244 (2004)
22. Wang, Y., Klijn, J.G., Zhang, Y., Sieuwerts, A.M., Look, M.P., Yang, F., Talantov, D., Timmermans, M., Meijer-van Gelder, M.E., Yu, J. et al.: Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer. Lancet **365**, 671–679 (2005)
23. Zhang, Z. and Zha, H.: Principal manifolds and non-linear dimension reduction via local tangent space aligment. SIAM Journal of Scientific Computing, **26** (1), 313–338 (2004)