Weighted Maximum Variance Dimensionality Reduction

Turki Turki[1,2] and Usman Roshan[2]

[1] Computer Science Department, King Abdulaziz University
P.O. Box 80221, Jeddah 21589, Saudi Arabia
tturki@kau.edu.sa
[2] Department of Computer Science, New Jersey Institute of Technology
University Heights, Newark, NJ 07102
usman@cs.njit.edu

Abstract. Dimensionality reduction procedures such as principal component analysis and the maximum margin criterion discriminant are special cases of a weighted maximum variance (WMV) approach. We present a simple two parameter version of WMV that we call 2P-WMV. We study the classification error given by the 1-nearest neighbor algorithm on features extracted by our and other dimensionality reduction methods on several real datasets. Our results show that our method yields the lowest average error across the datasets with statistical significance.

Keywords: dimensionality reduction, principal component analysis, maximum margin criterion

1 Introduction

The problem of dimensionality reduction arises in many data mining and machine learning tasks. Among many such algorithms the principal component analysis [1] (PCA) is a very popular choice. PCA seeks a vector $w \in R^d$ that solves

$$\arg\max_{w} \frac{1}{2n} \sum_{i,j} \frac{1}{n} (w^T(x_i - x_j))^2 \tag{1}$$

where $x_i \in R^d$ for $i = 0...n-1$. In other words it maximizes the variance of the projected data without taking class labels into consideration. The maximum margin criterion (MMC) [2] is a supervised dimensionality reduction method that overcomes limitations of the Fisher linear discriminant and has also shown to achieve higher classification accuracy [2]. It is given by w that maximizes $trace(w^T(S_b - S_w)w)$ subject to $w^T w = I$. Using Lagrange multipliers one can show that w is given by the largest eigenvectors of $S_b - S_w$.

In this paper we consider a general version of Equation 1 that we call the maximum weighted variance given by

$$\arg\max_w \frac{1}{2n} \sum_{i,j} C_{ij}(w^T(x_i - x_j))^2 \tag{2}$$

The above equation gives us both PCA and MMC for specific settings of C_{ij} as we show below. We consider a two parameter approach by setting $C_{ij} = \alpha < 0$ if x_i and x_j have the same class label and $C_{ij} = \beta > 0$ otherwise. In other words we simultaneously minimize the distance between projected pairwise points in the same class and maximize the same distance for points in different classes. For a given dataset we obtain α and β by 1-nearest neighbor cross-validation.

The straightforward eigendecomposition solution requires at least quadratic space in the dimensions of x_i. With graph Laplacians we can employ a singular value decomposition (SVD) approach to avoid this problem (as originally given in [3]) and thus apply it to high dimensional data. Below we describe our approach in detail followed by experimental results.

2 Methods

Suppose we are given the vectors $x_i \in R^d$ for $i = 0...n-1$ and a real matrix $C \in R^{n \times n}$. Let X be the matrix containing x_i as its columns (ordered x_0 through x_{n-1}). Now consider the optimization problem

$$\arg\max_w \frac{1}{2n} \sum_{i,j} C_{ij}(w^T(x_i - x_j))^2 \tag{3}$$

where $w \in R^d$ and C_{ij} is the entry in C corresponding to the i^{th} row and j^{th} column. This is in fact a more general representation of PCA and MMC.

2.1 Principal component analysis

To obtain PCA we set $C_{ij} = \frac{1}{n}$ and Equation 3 becomes (without the $\arg\max$ part)

$\frac{1}{2n}\sum_{i,j}\frac{1}{n}(w^T(x_i-x_j))^2 =$

$\frac{1}{2n}\sum_{i,j}\frac{1}{n}w^T(x_i-x_j)(x_i-x_j)^Tw =$

$\frac{1}{2n}\sum_{i,j}\frac{1}{n}w^T(x_ix_i^T-x_ix_j^T-x_jx_i^T+x_jx_j^T)w =$

$\frac{1}{2n}w^T\frac{1}{n}(\sum_{i,j}(x_ix_i^T-x_ix_j^T-x_jx_i^T+x_jx_j^T))w =$

$\frac{1}{2n}w^T\frac{1}{n}(2\sum_{i,j}x_ix_i^T-2\sum_{i,j}x_ix_j^T)w =$

$\frac{1}{2n}w^T\frac{1}{n}(2n\sum_{i}x_ix_i^T-2n^2mm^T)w =$

$\frac{1}{n}w^T(\sum_{i}x_ix_i^T-nmm^T)w =$

$w^T(\frac{1}{n}\sum_i(x_i-m)(x_i-m)^T)w =$

w^TS_tw

where $S_t=\frac{1}{n}\sum_i(x_i-m)(x_i-m)^T$ and is called the total scatter matrix. Inserting the optimization criterion into the last step yields $\arg\max_w w^TS_tw$ which is exactly the PCA optmization criterion [1].

2.2 Maximum margin discriminant

To obtain the MMC discriminant (a supervised learning method) first recall that the MMC optimization criterion is defined as $\arg\max_w w^T(S_b-S_w)w$ where S_b is the between-class scatter matrix and S_w is the within-class scatter matrix [2]. Since $S_b-S_w=S_t-2S_w$ where S_t is the total scatter matrix, this can be written as $\arg\max_w w^T(S_t-2S_w)w$ [4]. In practice though we would use the weighted maximum margin discriminant which is given by $\arg\max_w w^T(S_b-\alpha S_w)w$ [5]. We now set the weights C_{ij} to obtain this discriminant.

Suppose class labels $y_i\in\{+1,-1\}$ are provided for each x_i and n_k is the size of class k. Define C_{ij} to be $\frac{1}{n}$ if i and j have different class labels and $\frac{1}{n}-2\frac{1}{n_k}$ if i and j have the same class labels. We can then write Equation 3 as

$$\arg\max_w \frac{1}{2n}(\sum_{i,j}G_{ij}(w^T(x_i-x_j))^2-\sum_{i,j}2L_{ij}(w^T(x_i-x_j))^2) \tag{4}$$

where $G_{ij}=\frac{1}{n}$ for all i and j and $L_{ij}=\frac{1}{n_k}$ if i and j have class labels k and 0 otherwise. By substituting the values of G_{ij} and L_{ij} into Equation 4 and some symbolic manipulation we obtain the MMC discriminant

$$\tfrac{1}{2n}\textstyle\sum_{i,j} w^T(G_{ij}(x_i - x_j)(x_i - x_j) - 2L_{ij}(x_i - x_j)(x_i - x_j)^T)w =$$

$$\tfrac{1}{2n}(\textstyle\sum_{i,j} \tfrac{1}{n} w^T(x_i - x_j)(x_i - x_j)^T w-$$
$$2\textstyle\sum_{k=1}^{c} \sum_{cl(x_j)=k, cl(x_i)=k} \tfrac{1}{n_k} w^T(x_i - x_j)(x_i - x_j)^T w) =$$

$$\tfrac{1}{2n}(2\textstyle\sum_i^n w^T(x_i - m)(x_i - m)w-$$
$$2\textstyle\sum_{k=1}^{c} \tfrac{1}{n_k} \sum_{cl(x_j)=k, cl(x_i)=k} w^T(x_i x_i^T - x_i x_j^T - x_j x_i^T + x_j x_j^T)w) =$$

$$\tfrac{1}{2n}(2\textstyle\sum_i^n w^T(x_i - m)(x_i - m)w-$$
$$2\textstyle\sum_{k=1}^{c} \tfrac{1}{n_k} \sum_{cl(x_j)=k, cl(x_i)=k} w^T(2x_i x_i^T - 2x_i x_j^T)w) =$$

$$\tfrac{1}{2n}(2\textstyle\sum_i^n w^T(x_i - m)(x_i - m)w-$$
$$2\textstyle\sum_{k=1}^{c} \tfrac{1}{n_k} \sum_{cl(x_i)=k} w^T(2n_k x_i x_i^T - 2n_k^2 m_k m_k^T)w) =$$

$$\tfrac{1}{n}(\textstyle\sum_i^n w^T(x_i - m)(x_i - m)w-$$
$$2\textstyle\sum_{k=1}^{c} \sum_{cl(x_i)=k} w^T(x_i x_i^T - n_k m_k m_k^T)w) =$$

$$\tfrac{1}{n}(\textstyle\sum_i^n w^T(x_i - m)(x_i - m)w-$$
$$2\textstyle\sum_{k=1}^{c} \sum_{cl(x_i)=k} w^T(x_i - m_k)(x_i - m_k)^T)w) =$$

$$w^T(S_t - 2S_w)w$$

where m_k is the mean of points in k and $cl(x)$ returns the class of point x. The last equation in the above steps is just the MMC discriminant.

Equation 3 can be rewritten as $\arg\max_w \frac{1}{n} w^T X L X^T w$ where $L = D - C$ and $D_{ii} = \sum_i C_{ii}$ [6]. The matrix L is called the Laplacian of the weight matrix C. Using Lagrange multipliers one can show that the largest eigenvector of $\frac{1}{n} X L X^T$ (i.e. eigenvector with largest eigenvalue) is the solution to w [6]. Thus, the largest eigenvector is also the solution to PCA and MMC.

2.3 Laplacian linear discriminant analysis

Following the Laplacian framework we can write the MMC discriminant (Equation 4) as $\arg\max_w \frac{1}{n} w^T X(L_g - 2L_l)X^T w$ where L_g is the Laplacian of G and L_l is the Laplacian of L [3,4]. This form of the the maximum margin discriminant is also called Laplacian linear discriminant analysis and has been studied for unsupervlsed learning [4]. As in PCA and MMC the largest eigenvector of $\frac{1}{n} X(L_g - 2L_l)X^T$ is the solution to the Laplacian discriminant.

Notice that C_{ij} in Equation 3 can take on arbitrary values. With suitable settings we obtained PCA and MMC. How does one select the best values C_{ij} for a particular problem? Our solution is to collapse values of C into two parameters and select their values that minimize error on the training data.

2.4 Two parameter weighted maximum variance discriminant

As shown above the MMC discriminant is obtained by setting $G_{ij} = \frac{1}{n}$ for all i and j and $L_{ij} = \frac{1}{n_k}$ if i and j have class labels k and 0 otherwise in Equation 4. We consider a different setting for L below which gives us the two parameter weighted maximum variance discriminant (2P-WMV). We also show that this yields a class-wise unnormalized within-class scatter matrix and a pairwise inter-class scatter matrix.

Define the matrix $G \in R^{n \times n}$ as $G_{ij} = \frac{1}{n}$ for all i and j and $L \in R^{n \times n}$ as

$$L_{ij} = \begin{cases} \alpha & \text{if } y_i = y_j \\ \beta & \text{if } y_i \neq y_j \\ 0 \text{ if } y_i \text{ or } y_j \text{ is undefined} \end{cases}$$

Substituting these values into Equation 4 we obtain

$$\begin{array}{l} \frac{1}{2n}(\sum_{i,j}\frac{1}{n}w^T(x_i - x_j)(x_i - x_j)w \\ -2\sum_{cl(x_i)=cl(x_j)} \alpha w^T(x_i - x_j)(x_i - x_j)^T w \\ -2\sum_{cl(x_i)\neq cl(x_j)} \beta w^T(x_i - x_j)(x_i - x_j)^T w) = \\ \\ \frac{1}{2n}(2\sum_i^n w^T(x_i - m)(x_i - m)w \\ -2\sum_{k=1}^c \alpha 2n_k \sum_{cl(x_j)=k} w^T(x_j - m_k)(x_j - m_k)^T w \\ -2\beta\sum_{c=1}^k \sum_{d=c+1}^k \sum_{cl(x_i)=c,cl(x_j)=d} w^T(x_i - x_j)(x_i - x_j)^T w) = \\ \\ \frac{1}{n}\sum_i^n w^T(x_i - m)(x_i - m)w \\ -2\alpha\frac{1}{n}\sum_{k=1}^c n_k \sum_{cl(x_j)=k} w^T(x_j - m_k)(x_j - m_k)^T w \\ -2\beta\frac{1}{n}\sum_{c=1}^k \sum_{d=c+1}^k \sum_{cl(x_i)=c,cl(x_j)=d} w^T(x_i - x_j)(x_i - x_j)^T w = \\ \\ w^T S_t w - 2(\alpha w^T S'_w w + \beta w^T S'_b w) = \\ \\ w^T(S_t - 2(\alpha S'_w + \beta S'_b))w \end{array}$$

where

$$\begin{array}{l} S'_w = \frac{1}{n}\sum_{k=1}^c n_k \sum_{cl(x_j)=k}(x_j - m_k)(x_j - m_k)^T \\ S'_b = \frac{1}{2n}\sum_{c=1}^k \sum_{d=c+1}^k \sum_{cl(x_i)=c,cl(x_j)=d}(x_i - x_j)(x_i - x_j)^T \end{array}$$

Note the similarity of S'_w to the standard within-class matrix used in MMC given by $S_w = \frac{1}{n}\sum_i^k \sum_{cl(x_j)=i}(x_j - m^i)(x_j - m^i)^T$. S_w is the class-wise normalized version of S'_w. Thus, the discriminant yielded by our approach is given by the standard total scatter matrix, a modified within-class matrix, and a pairwise inter-class scatter matrix. We can obtain MMC by setting $\alpha = \frac{1}{n_k}$ if $y_i = k, y_j = k$ and $\beta = 0$. This discards the inter-class scatter matrix and makes $S'_w = S_w$.

After defining L and G compute L_g the Laplacian of G, L_l the Laplacian of L, and the matrix $\frac{1}{n}X(L_g - L_l)X^T$ (the 2P-WMV discriminant). The solution to 2P-WMV is w that maximizes $\frac{1}{n}w^T X(L_g - L_l)X^T w$ which is in turn is given by the largest eigenvector of $\frac{1}{n}X(L_g - L_l)X^T$ [4].

3 Results

To evaluate the classification ability of our extracted features we use the simple and popular 1-nearest neighbor (1NN) algorithm. In 10-fold and 5-fold cross-validation experiments we apply the 1-nearest neighbor classification algorithm to features extracted from our method 2P-WMV, the weighted maximum margin discriminant (WMMC), PCA, and the features as they are (denoted simply as 1NN). We calculate average error rates across 50 randomly selected datasets shown in Table 1 from the UCI Machine Learning Repository [7].

3.1 Experimental methodology

We compare four classification algorithms: 2P-WMV+1NN, PCA+1NN, WMMC+1NN, and 1NN where the first three are 1NN applied to features extracted from each of the three dimensionality reduction algorithms. We use 10-fold cross-validation on each real dataset with the same set of splits for each algorithm. However, for datasets with fewer than 300 instances we use 5-fold cross-validation to obtain a large enough validation set. For dimensionality reduction we find the best parameters and number of dimensions by cross-validating further on the training dataset (also 10-fold).

In 2P-WMV we let β range from {-2,-1.9,-1.8,-1.7,-1.6,-1.5,-1.4,-1.3,-1.2,-1.1,-1,-.9,-.8,-.7,-.6,-.5,-.4,-.3,-.2,-.1,-.01} and α fixed to 1. For WMMC we let the α parameter range from $\{10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 100\}$. Recall that WMMC is given by $\arg\max_w w^T(S_b - \alpha S_w)w$ [5]. For each parameter we reduce dimensionality to 20 and then pick the top $1 <= k <= 20$ features that give the lowest 1NN error on the training. Thus the cross-validation on the training set gives us the best values of α and the reduced number of features (including PCA) which we then apply to the validation set.

We wrote our code in C and R and make it freely available at http://www.cs.njit.edu/usman/wmv/. Our C programs use CLAPACK libraries for performing the eigenvector and singular value decompositions.

3.2 Experimental results on fifty datasets

We compute the $\frac{number\_of\_misclassifications}{number\_of\_instances\_in\_validation}$ for each training-validation split during cross-validation and take the mean to be the average cross-validation error. In Table 2 we show the average cross-validation error on each dataset. Across the the 50 datasets 2P-WMV+1NN achieves the lowest average error of 13.324% and has the lowest error in 21 out of the 50 datasets. The next best is WMMC+1NN that achieves an average error of 15.302% and has the lowest error in 12 out of the 50 datasets. PCA+1NN are 1NN have higher average errors

Code	Dataset	Classes	Dimension	Instances
1	Liver-disorders	2	6	345
2	Diabetes	2	8	768
3	Breast Cancer	2	10	683
4	Page block	5	10	5473
5	Wine-quality-red	11	11	1599
6	Wine quality	11	11	4898
7	Wine	3	13	178
8	Heart	2	13	270
9	Australian Credit Approval	2	14	690
10	EEG Eye State	2	14	14980
11	Pen-Based Recognition	10	16	10992
12	Climate	2	18	540
13	lymphography	4	18	148
14	Statlog image	7	19	2310
15	Two norm	2	20	7400
16	Ring	2	20	7400
17	Cardiotocography	10	21	2126
18	Thyroid	3	21	7200
19	Waveform	3	21	5000
20	Statlog German credit card	2	24	1000
21	Steel faults	7	27	1941
22	Breast cancer	2	30	569
23	Ionosphere	2	34	351
24	Dermatology	6	34	366
25	Statlog	7	36	6435
26	Texture	11	40	5500
27	Waveform	3	40	5000
28	Qsar	2	41	1055
29	SPECTF heart	2	44	267
30	Mlprove	6	51	6118
31	Spambase	2	57	4597
32	Sonar	2	60	208
33	Digits	2	63	762
34	Ozone	2	72	1847
35	Insurance company coil2000	2	85	5822
36	Movement libras	15	90	360
37	Hill valley	2	100	606
38	BCI	2	117	400
39	Gas sensor array drift	6	128	13910
40	Musk	2	166	476
41	Coil	6	241	1500
42	Scene classification	6	294	2230
43	Madelon	2	500	2600
44	Smartphone	6	561	10299
45	Secom	2	591	1567
46	Mfeat	10	649	2000
47	CNAE-9	9	857	1080
48	ACASVA actions	2	960	11288
49	Micromass	2	1300	931
50	Gisette	2	5000	1000

Table 1: Datasets that from the UCI Machine Learning repository that we used in our study [7]

Code	Dataset	2P-WMV+1NN	WMMC+1NN	PCA+1NN	1NN
1	Liver-disorders	**0.364**	0.376	0.4	0.404
2	Diabetes	0.31912	0.33382	0.34706	0.31912
3	Breast Cancer	**0.03016**	0.03492	0.37937	0.37937
4	Page block	0.04586	**0.04199**	0.04622	0.04622
5	Wine-quality-red	**0.37718**	0.37785	0.42081	0.42013
6	Wine quality	**0.37582**	0.38381	0.4043	0.40451
7	Wine	0.075	0.075	0.2125	0.2125
8	Heart	**0.21**	0.33	0.425	0.42
9	Australian Credit Approval	**0.20833**	0.21667	0.44167	0.43167
10	EEG Eye State	**0.0198**	0.02094	0.0202	0.0202
11	Pen-Based Recognition	0.00586	0.00614	0.00577	0.00577
12	Climate	**0.066**	0.072	0.14	0.132
13	Lymphography	0.2	0.21786	0.21429	0.2
14	Statlog image	0.03609	**0.03435**	0.03609	0.03565
15	Two norm	**0.0289**	0.02918	0.03342	0.05315
16	Ring	0.14685	**0.14014**	0.15425	0.24274
17	Cardiotocography	0.08398	0.08932	0.08495	**0.0835**
18	Thyroid	**0.03915**	0.06211	0.07014	0.07014
19	Waveform	0.18143	**0.18**	0.18612	0.22857
20	Statlog German credit card	**0.33444**	0.37	0.35667	0.35444
21	Steel faults	0.36126	**0.36073**	0.61885	0.61885
22	Breast cancer	**0.07755**	0.11429	0.09388	0.09388
23	Ionosphere	0.06452	**0.05806**	0.10323	0.10968
24	Dermatology	**0.01538**	0.03462	0.11538	0.11538
25	Statlog	**0.09118**	0.11874	0.09496	0.09512
26	Texture	0.00926	0.01315	0.00944	**0.00796**
27	Waveform	0.18143	0.18755	**0.17898**	0.23837
28	Qsar	0.18211	**0.16211**	0.20316	0.19895
29	SPECTF heart	0.27647	**0.25882**	0.28235	0.26471
30	Mlprove	0.42204	0.44128	0.41941	**0.41382**
31	Spambase	0.08709	**0.08249**	0.17221	0.16565
32	Sonar	0.17222	0.2	0.15556	0.15556
33	Digits	0.01111	0.01806	0.01111	**0.00972**
34	Ozone	0.10904	**0.09718**	0.10678	0.10565
35	Insurance company coil2000	0.1042	0.10262	**0.0965**	0.09685
36	Movement libras	0.10333	0.12333	0.10333	**0.09667**
37	Hill valley	**0.02321**	0.06429	0.41607	0.42143
38	BCI	**0.16333**	0.17667	0.44667	0.41333
39	Gas sensor array drift	0.00878	0.01058	0.00878	0.00885
40	Musk	**0.11957**	0.23696	0.13478	0.1587
41	Coil	0.02286	0.03429	0.02143	**0.01429**
42	Scene classification	0.29454	0.335	0.29636	**0.28909**
43	Madelon	**0.1256**	0.4568	0.1268	0.3444
44	Smartphone	0.04563	0.04194	0.07363	**0.02623**
45	Secom	**0.08027**	0.11429	0.1	0.10204
46	Mfeat	0.05526	**0.05158**	0.05211	0.05263
47	CNAE-9	0.069	**0.065**	0.176	0.132
48	ACASVA actions	**0.11637**	0.18479	0.17809	0.1178
49	Micromass	0.07253	0.06264	0.11209	**0.05934**
50	Gisette	**0.04889**	0.05111	0.09556	0.08222
	Average error	0.13324	0.15302	0.18765	0.18946

Table 2: Average cross-validation error of different algorithms on each of the 50 real datasets from the UCI machine learning repository. Shown in bold is the method with the lowest unique error.

at 18.765% and 18.946% respectively. PCA+1NN and 1NN have the lowest error in 2 and 9 out of the 50 datasets respectively.

We measure the statistical significance with the Wilcoxon rank test [9]. This is a standard test to measure the difference between two methods across a number of datasets. Roughly speaking it shows statistical significance between two methods when one outperforms the other each time on a large number of datasets. In Table 3 the p-values show that 2P-WMV+1NN statistically significantly outperforms the other three method across all 50 datasets.

	WMMC+1NN	PCA+1NN	1NN
2P-WMV+1NN	.0004	$< .0001$	.0001
WMMC+1NN		.0232	.0536
PCA+1NN			.0949

Table 3: Wilcox rank test p-values (two-tailed test) between all pairs of methods.

4 Discussion

Both 2PWMV+1NN and WMMC+1NN reduce dimensionality by determining optimal parameters specific to the given dataset. This approach is better than the unsupervised PCA and the non-parametric MMC (results not shown here). In fact 1NN applied to the raw data can be better than non-parameteric MMC most of the time.

In this study we fixed α for 2PWMV and varied only β. If we cross-validated α we could potentially obtain lower error but at the cost of increased running time. In the current experiments 2PWMV+1NN and WMMC+1NN are the slowest methods yet still tractable for large datasets.

We chose 1NN as the classification method for this study due to its simplicity and its popularity with dimensionality reduction programs. Other classifiers such as the support vector machine [1] may perform better when replaced with 1NN. However, in that case the regularization parameter would also need to be optimized via cross-validation which increases the total runtime.

5 Conclusion

We introduce a two parameter variant of the weighted maximum variance discriminant and optimize it with cross-validation followed by 1-nearest neighbor for classification. Compared to existing approaches our method obtains the lowest average error with statistical significance across several real datasets from the UCI machine learning repository.

References

1. Alpaydin, E.: Machine Learning. MIT Press (2004)
2. Li, H., Jiang, T., Zhang, K.: Efficient and robust feature extraction by maximum margin criterion. In Thrun, S., Saul, L., Schölkopf, B., eds.: Advances in Neural Information Processing Systems 16. MIT Press, Cambridge, MA (2004)
3. Tang, H., Fang, T., Shi, P.F.: Rapid and brief communication: Laplacian linear discriminant analysis. Pattern Recogn. **39**(1) (2006) 136–139
4. Niijima, S., Okuno, Y.: Laplacian linear discriminant analysis approach to unsupervised feature selection. Computational Biology and Bioinformatics, IEEE/ACM Transactions on **6**(4) (2009) 605–614
5. Zheng, W., Zou, C., Zhao, L.: Weighted maximum margin discriminant analysis with kernels. Neurocomputing **67** (2005) 357–362
6. He, X., Niyogi, P.: Locality preserving projections. In Thrun, S., Saul, L., Schölkopf, B., eds.: Advances in Neural Information Processing Systems 16. MIT Press, Cambridge, MA (2004)
7. A. Asuncion, D.N.: UCI machine learning repository (2007)
8. Guyon, I., Gunn, S., Ben-Hur, A., Dror, G.: Result analysis of the nips 2003 feature selection challenge. In: Advances in Neural Information Processing Systems. (2004) 545–552
9. Kanji, G.K.: 100 Statistical Tests. Sage Publications Ltd (1999)