Int. J. Mach. Learn. & Cyber.
DOI 10.1007/s13042-012-0131-7

ORIGINAL ARTICLE

Principal component analysis using QR decomposition

Alok Sharma · Kuldip K. Paliwal · Seiya Imoto · Satoru Miyano

Received: 31 March 2012 / Accepted: 3 September 2012

Abstract In this paper we present QR based principal component analysis (PCA) method. Similar to the singular value decomposition (SVD) based PCA method this method is numerically stable. We have carried out analytical comparison as well as numerical comparison (on Matlab software) to investigate the performance (in terms of computational complexity) of our method. The computational complexity of SVD based PCA is around $14dn^2$ flops (where d is the dimensionality of feature space and n is the number of training feature vectors); whereas the computational complexity of QR based PCA is around $2dn^2 + 2dth$ flops (where t is the rank of data covariance matrix and h is the dimensionality of reduced feature space). It is observed that the QR based PCA is more efficient in terms of computational complexity.

Keywords Principal component analysis · Singular value decomposition · QR decomposition · Computational complexity

A. Sharma (✉) · S. Imoto · S. Miyano
Laboratory of DNA Information Analysis, Human Genome Center, Institute of Medical Science, University of Tokyo, 4-6-1 Shirokanedai, Minato-ku, Tokyo 108-8639, Japan
e-mail: sharma_al@usp.ac.fj

A. Sharma
School of Engineering and Physics, University of the South Pacific, Suva, Fiji

K. K. Paliwal
Signal Processing Lab, School of Engineering, Griffith University, Nathan, Australia

1 Introduction

Principal component analysis (PCA) is an important technique used for dimensionality reduction and has been applied in pattern classification and data representation areas. It works on training data $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$ in a non-supervised manner, where d is the dimensionality of the data. It does not require class labels for individual feature vectors x_j. In conventional PCA procedure, a covariance matrix $\mathbf{\Sigma_x} = \mathbf{H}\mathbf{H}^{\mathrm{T}}$ (where $\mathbf{H} = \frac{1}{\sqrt{n}}[\mathbf{x}_1 - \boldsymbol{\mu}, \mathbf{x}_2 - \boldsymbol{\mu}, \ldots, \mathbf{x}_n - \boldsymbol{\mu}]$ and $\boldsymbol{\mu} = \frac{1}{\mathrm{n}}\sum_{j=1}^{n} \mathbf{x}_j$ is the centroid of training data) is formed and its eigenvalue decomposition (EVD) is done to extract $h \leq t$ (where $t = rank(\mathbf{H})$) eigenvectors corresponding to h leading eigenvalues (i.e., those eigenvectors with the largest associated eigenvalues of $\mathbf{\Sigma_x}$ or $\mathbf{H}$). The value of h is between $[1, t]$ which is representing the dimensionality of the reduced dimensional space.

In many applications occurring in face recognition and biometrics areas, the dimensionality d is larger than the number of training samples n; these come under the commonly used title, 'the small sample size problem' [2, 7, 9, 13, 16–19, 21, 22]. In such a problem, the covariance matrix $\mathbf{\Sigma_x}$ will be of size $d \times d$ and it will be a slow procedure (much computer power needed) to compute EVD of this matrix. However, a fast procedure to carry out EVD is given in Fukunaga [5] and has been extensively applied in pattern recognition literature. Some other procedures which are fast but limited in accuracy for obtaining the eigenvectors are also used in some applications [3, 4, 10–12, 14, 15]. As explained in Fukunaga [5], EVD of $\mathbf{H}^{\mathrm{T}}\mathbf{H}$ is carried out to find the eigenvalues and eigenvectors of $\mathbf{\Sigma_x} = \mathbf{H}\mathbf{H}^{\mathrm{T}}$. However, this method has a problem that it can be numerically unstable. This is because the formation of $\mathbf{H}^{\mathrm{T}}\mathbf{H}$ matrix may lose precision (as shown in [6], pg. 239);

Published online: 25 September 2012

since the hardware has a fixed arithmetic, it is possible that when the square matrix $\mathbf{H}^{\mathrm{T}}\mathbf{H}$ is formed a small decimal value in the rectangular matrix $\mathbf{H}$ becomes squared thereby its value will be below the precision limit of the hardware. This has an implication that the formation of the matrix will lose a rank or otherwise the information. Due to the above two reasons (computational complexity and precision), the eigenvalue analysis of a large size covariance matrix $\mathbf{\Sigma_x} = \mathbf{H}\mathbf{H}^{\mathrm{T}}$ is not done, instead singular value decomposition (SVD) is directly applied on rectangular matrix $\mathbf{H}$ [20]. The computational complexity of this SVD based method is, however, still high (approximately $14dn^2$ flops). In the chemometrics literature, a faster way of computing PCA has been proposed, however, with limited accuracy. For example postponed basis matrix multiplication(PBM)-PCA method has been proposed [1, 8]. This strategy does not handle the data matrix $\mathbf{X}$ directly, however, it assumes that data matrix $\mathbf{X}$ is first decomposed into three matrices: $\mathbf{C}$, $\mathbf{B}_1$ and $\mathbf{B}_2$, where $\mathbf{X} \approx \mathbf{B}_1\mathbf{C}\mathbf{B}_2^{\mathbf{T}}$. Matrices $\mathbf{B}_1$ and $\mathbf{B}_2$ are B-spline basis sets. Therefore, prior computation is required to obtain these matrices. Secondly, $\mathbf{B}_1\mathbf{C}\mathbf{B}_2^{\mathbf{T}}$ is just the approximation of actual data matrix $\mathbf{X}$ and thus it is not very accurate[1]. In the present paper, our objective is to derive computationally efficient as well as numerically stable PCA method. In order to make it computationally stable we use matrix $\mathbf{H}$; and to make it computationally efficient, we use QR decomposition. The QR based PCA method approximately requires only $2\,dn^2 + 2\,dh^2$ flops. Thus, the proposed method is numerically stable and furthermore, it is more efficient (in terms of computational complexity) than the SVD based PCA method.

2 Principal component analysis

The PCA transform $\mathbf{\Phi} \in \mathbb{R}^{d\times h}$ is used in transforming d-dimensional feature vectors to h-dimensional feature vectors, where $h < d$. It can also be used to reconstruct d-dimensional feature vectors back from h-dimensional feature vectors with some finite error known as the reconstruction error. In PCA, this reconstruction error is minimum in the mean square error (MSE) sense. The transformation matrix $\mathbf{\Phi}$ that minimizes the MSE satisfies $\mathbf{\Sigma_x}\boldsymbol{\phi}_i = \lambda_i\boldsymbol{\phi}_i$, where $\mathbf{\Sigma_x} \in \mathbb{R}^{d\times d}$ is the covariance matrix, λ_i are the eigenvalues and $\boldsymbol{\phi}_i \in \mathbb{R}^{d\times 1}$ are eigenvectors corresponding to λ_i (for $i = 1\ldots h$). The column vectors of $\mathbf{\Phi}$ are the leading eigenvectors $\boldsymbol{\phi}_i$; i.e., these eigenvectors correspond to the largest eigenvalues.

The covariance matrix $\mathbf{\Sigma_x}$ is a symmetric matrix which can be expressed as $\mathbf{\Sigma_x} = \mathbf{H}\mathbf{H}^{\mathrm{T}}$. If the dimensionality is extremely large ($d \gg n$), then the computation of EVD of $\mathbf{\Sigma_x} = \mathbf{H}\mathbf{H}^{\mathrm{T}}$ becomes a slow procedure. In this case an economical way would be to compute EVD of $\mathbf{H}^{\mathrm{T}}\mathbf{H}$ instead of $\mathbf{H}\mathbf{H}^{\mathrm{T}}$. However, the formation of $\mathbf{H}^{\mathrm{T}}\mathbf{H}$ could lead to a loss of information [6]. The more accurate and reliable way would be to use SVD of $\mathbf{H}$ which will give eigenvectors and square root eigenvalues of $\mathbf{\Sigma_x}$. This requires around $14dn^2 - 2n^3$ flops. In many applications, a faster yet accurate implementation is desired. In the next section, the QR decomposition based PCA is described which can extract eigenvectors and square root eigenvalues of $\mathbf{\Sigma_x}$ in a numerically stable manner, similar to the SVD based PCA method. However, its computational complexity is much lower than the SVD based PCA method.

3 QR based PCA method

In this section, QR based PCA method is introduced. This method uses the rectangular matrix $\mathbf{H} \in \mathbb{R}^{d\times n}$ (where $d \gg n$) to carry out EVD of $\mathbf{\Sigma_x} = \mathbf{H}\mathbf{H}^{\mathrm{T}}$ in a numerically stable manner. Let the rank of $\mathbf{\Sigma_x} \in \mathbb{R}^{d\times d}$ be t, where $1 \le t < n$. The rectangular matrix $\mathbf{H}$ can be decomposed into orthogonal matrix $\mathbf{Q}_1 \in \mathbb{R}^{d\times t}$ and upper triangular matrix $\mathbf{R}_1 \in \mathbb{R}^{t\times n}$ using economic QR decomposition as

$$\mathbf{H} = \mathbf{Q}_1\mathbf{R}_1 \tag{1}$$

Substituting Eq. 1 in $\mathbf{\Sigma_x} = \mathbf{H}\mathbf{H}^{\mathrm{T}}$, we get

$$\mathbf{\Sigma_x} = \mathbf{Q}_1\mathbf{R}_1\mathbf{R}_1^{\mathrm{T}}\mathbf{Q}_1^{\mathrm{T}} \tag{2}$$

The matrix $\mathbf{R}_1^{\mathrm{T}}$ can be factored by SVD as

$$\mathbf{R}_1^{\mathrm{T}} = \mathbf{U}_1\mathbf{D}_1\mathbf{V}^{\mathrm{T}} \tag{3}$$

where $\mathbf{U}_1 \in \mathbb{R}^{n\times t}$ and $\mathbf{V} \in \mathbb{R}^{t\times t}$ are orthogonal matrices and $\mathbf{D}_1 \in \mathbb{R}^{t\times t}$ is a diagonal matrix. Substituting Eq. 3 in Eq. 2, we get

$$\begin{aligned}
&\mathbf{\Sigma_x} = \mathbf{Q}_1\mathbf{V}\mathbf{D}_1\mathbf{U}_1^{\mathrm{T}}\mathbf{U}_1\mathbf{D}_1\mathbf{V}^{\mathrm{T}}\mathbf{Q}_1^{\mathrm{T}}\\
&\text{or}\quad \mathbf{\Sigma_x} = \mathbf{Q}_1\mathbf{V}\mathbf{D}_1^2\mathbf{V}^{\mathrm{T}}\mathbf{Q}_1^{\mathrm{T}}\\
&\text{or}\,\mathbf{\Sigma_x} = \mathbf{Q}_1\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{\mathrm{T}}\mathbf{Q}_1^{\mathrm{T}}
\end{aligned}$$

where $\mathbf{\Lambda} = \mathbf{D}_1^2$. Since $(\mathbf{Q}_1\mathbf{V})^{\mathrm{T}}(\mathbf{Q}_1\mathbf{V}) = \mathbf{V}^{\mathrm{T}}\mathbf{Q}_1^{\mathrm{T}}\mathbf{Q}_1\mathbf{V} = \mathbf{V}^{\mathrm{T}}(\mathbf{I})\mathbf{V} = \mathbf{I}$, the transform $\mathbf{Q}_1\mathbf{V}$ is an orthogonal matrix. This orthogonal transform also diagonalizes matrix $\mathbf{\Sigma_x}$; i.e., transform $\mathbf{Q}_1\mathbf{V}$ is indeed an eigenvector matrix and $\mathbf{\Lambda}$ is an eigenvalue matrix of $\mathbf{\Sigma_x}$.

Let $\mathbf{V}_h \in \mathbb{R}^{t\times h}$ (where, $1 \le h \le t$) be a matrix containing h column vectors of $\mathbf{V}$ corresponding to the largest h

[1] To deal with higher order matrices, other variants of PCA have been proposed in the literature. See the following references for details [23–27]

Table 1 QR based PCA method Algorithm

Step 1. Compute $\mathbf{Q}_1 \in \mathbb{R}^{d\times t}$ and $R \in \mathbb{R}^{t\times n}$ using economic QR decomposition of matrix $\mathbf{H} \in \mathbb{R}^{d\times n}$, where n is the number of samples, d is the dimension and $t = rank(\mathbf{H})$ with $1 \leq t < n$

Step 2. Using economic SVD of $\mathbf{R}_1^{\mathrm{T}}$ matrix, compute diagonal matrix $\mathbf{D}_1 \in \mathbb{R}^{t\times t}$ and orthogonal matrix $\mathbf{V} \in \mathbb{R}^{t\times t}$ (Note that economic SVD of $\mathbf{R}_1$ can also be used to give the orthogonal matrix $\widehat{\mathbf{U}}_1 \in \mathbb{R}^{t\times t}$ (which can be used in place of $\mathbf{V}$) and diagonal matrix $\mathbf{D}_1 \in \mathbb{R}^{t\times t}$).

Step 3. Find $\mathbf{V}_h \in \mathbb{R}^{t\times h}$, the h column vectors of $\mathbf{V}$ that correspond to the h largest diagonal entries of $\mathbf{D}_1$.

Step 4. Compute PCA transform $\mathbf{\Phi} = \mathbf{Q}_1\mathbf{V}_h$.

Table 2 Computation complexity of QR based PCA method

Significant processing steps of the method	Computational complexity
Economic QR decomposition of $\mathbf{H} \in \mathbb{R}^{d\times n}$	$2dn^2 - 2n^3/3$
SVD procedure to find $\mathbf{D}_1$ and $\mathbf{V}$	$4nt^2 + 8t^3$
Multiplication of $\mathbf{Q}_1$ and $\mathbf{V}_h$	$2dth$
Total estimated	$2dn^2 + 2dth + 4nt^2 + 8t^3 - 2n^3/3$

diagonal entries of $\mathbf{D}_1$, then the PCA transform would be $\mathbf{\Phi} = \mathbf{Q}_1\mathbf{V}_h \in \mathbb{R}^{d\times h}$. The QR based PCA method is summarized in Table 1.

The computational complexity of QR based PCA method can be described as follows. The economic QR decomposition on rectangular matrix $\mathbf{H} \in \mathbb{R}^{d\times n}$ requires $2dn^2$ flops (if modified Gram-Schmidt QR method is used) or $2dn^2 - 2n^3/3$ flops (if fast Givens QR method or Householder QR method is used) [6]. The economic SVD of $\mathbf{R}_1^{\mathrm{T}}$ to find diagonal matrix $\mathbf{D}_1 \in \mathbb{R}^{t\times t}$ and orthogonal matrix $\mathbf{V} \in \mathbb{R}^{t\times t}$ requires $4nt^2 + 8t^3$ flops [6]. Finally the multiplication of $\mathbf{Q}_1$ and $\mathbf{V}_h$ requires $2dth$ flops. The summary of the computational complexity of the method is given in Table 2.

From Table 2, it can be observed that if the dimensionality d is very large compared to the number of samples n (i.e., $d \gg n$), then the total estimated computational complexity would be $2dn^2 + 2dth$ flops. In a special case if only one dominant eigenvector is required then computational complexity would be around $2d(n^2 + t)$.

4 Experimentation

The QR based as well as SVD based PCA methods use $\mathbf{H}$ to compute EVD of $\mathbf{\Sigma_x} = \mathbf{H}\mathbf{H}^{\mathrm{T}}$. Thus, they are both numerically stable. In this section, we compare these methods in terms of their computational complexities. For this, we generate data with 3,000 samples and variable dimension from 5,000 to 10,000. The rank of the data is set to be 1,000. The value of h is set to be 10. Figure 1a shows the analytical comparison of flops between QR based PCA and SVD based PCA methods. Figure 1b shows the numerical comparison of their cpu times when implemented on Matlab software. It can be seen from this figure that the proposed method is computationally more efficient

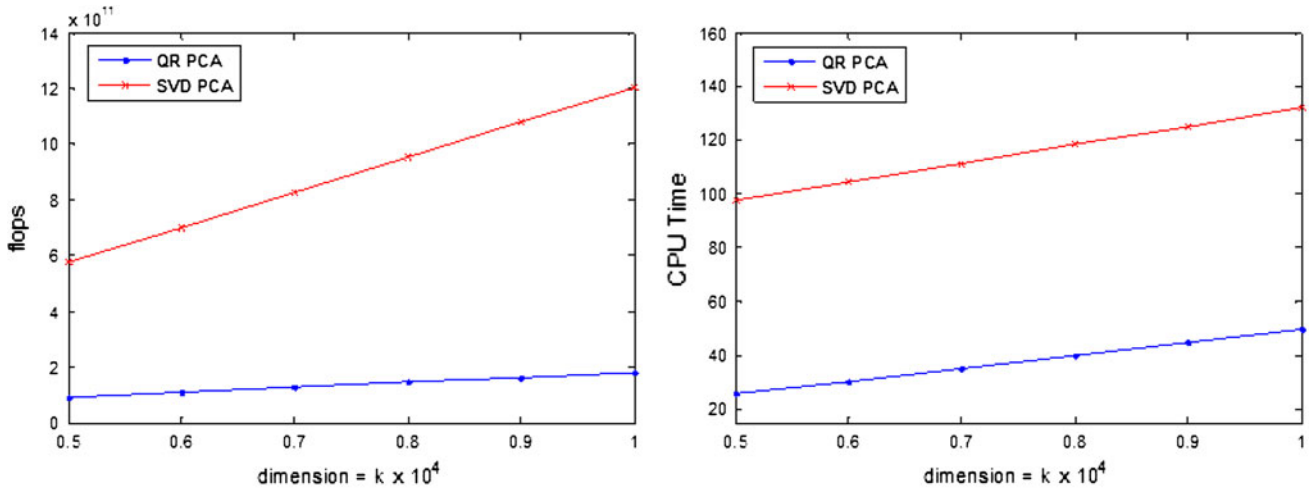

Fig. 1 A comparison of cpu time between QR based PCA and SVD based PCA methods

than the SVD based PCA method both analytically and Matlab-wise. In addition, the ratio of cpu times of SVD based PCA method and QR based PCA method is close to the ratio of their theoretical determinations of flops.

The computational advantage of the proposed method mainly comes from the utilization of QR decomposition of covariance matrix. The QR decomposition procedure decomposes rectangular matrix $\mathbf{H}$ into orthogonal matrix and upper triangular matrix. Thereafter, SVD procedure can be applied to the upper triangular matrix to compute leading eigenvectors. This is a cost efficient procedure compared to using SVD directly on the rectangular matrix $\mathbf{H}$. The computational effectiveness of the QR based PCA method compared with SVD based PCA method would be significant especially for low rank rectangular matrix $\mathbf{H}$.

5 Conclusion

A QR based PCA method has been presented in this paper. Like the SVD based PCA method, this method uses $\mathbf{H}$ for EVD of $\mathbf{\Sigma_x} = \mathbf{H}\mathbf{H}^{\mathrm{T}}$ for numerical stability. It is shown in this paper that the proposed method outperforms the SVD based PCA method in terms of computational complexity.

Since a computationally efficient way of doing dimensionality reduction is crucial in many fields of research, a number of applications of QR based PCA method can be envisaged. For instance, it can be applied to face recognition problem [28, 29], attribute reduction [30], decision tree induction [31, 32] and biometrics applications [33, 34].

Acknowledgments We thank the Reviewers and the Editor for their constructive comments which appreciably improved the presentation quality of the paper.

References

1. Alsberg BK, Kvalheim OM (1994) Speed improvement of multivariate algorithms by the method of postponed basis matrix multiplication Part I. Principal component analysis. Chemom Intell Lab Syst 24:31–42
2. Belhumeur PN, Hespanha JP, Kriegman DJ (1997) Eigenfaces vs. fisherfaces: recognition using class specific linear projection. IEEE T PAMI 19(7):711–720
3. Chen T-C, Chen K, Liu W, Chen L-G (2009) One-chip principal component analysis with a mean pre-estimation method for spike sorting. In: Proceedings of IEEE international symposium on circuits and systems, pp 3310–3113
4. Chen T-C, Liu W, Chen L-G (2008) VLSI architecture of leading eigenvector generation for on-chip principal component analysis spike sorting system. In: Proceedings of 30th Annual International Conference IEEE engineering in medicine and biology society EMBS'08, pp 3192–3195
5. Fukunaga K (1990) Introduction to Statistical Pattern Recognition. Academic Press, USA
6. Golub GH, Loan CFV (1996) Matrix Computations. The John Hopkins University Press, Baltimore
7. Jian H, Yuen PC, Wen-Sheng C (2004) A novel subspace LDA algorithms for recognition of face images with illumination and pose variations. Proc Int Conf Mach Learn Cybern 6:3589–3594
8. Kiers HAL, Harshman RA (1997) Relating two proposed methods for speed up of algorithms for fitting two- and three- way principal component and related multilinear models. Chemom Intell Lab Syst 36:31–40
9. Paliwal KK, Sharma A (2010) Improved direct LDA and its application to DNA microarray gene expression data. Pattern Recogn Lett 31:2489–2492
10. Roweis S (1997) EM Algorithms for PCA and SPCA. Neural Inf Process Syst 10:626–632
11. Sajid I, Ahmed MM, Taj I (2008) Design and implementation of a face recognition system using fast PCA. In: Proceedings of the International Symposium on Computer Science and its Applications CSA, pp 126–130
12. Schilling RJ, Harris SL (2000) Applied Numerical Methods for Engineers Using Matlab and C. Brooks/Cole Publishing, Boston
13. Sharma A, Paliwal KK (2010) Regularisation of eigenfeatures by extrapolation of scatter-matrix in face-recognition problem. Electron Lett IEE 46(10):682–683
14. Sharma A, Paliwal KK (2007) Fast principal component analysis using fixed-point algorithm. Pattern Recogn Lett 28(10):1151–1155
15. Sirovich L (1987) Turbulence and the dynamics of coherent structures. Quart Appl Math XLV 45:561–571
16. Song F, Zhang D, Wang J, Liu H, Tao Q (2007) A parameterized direct LDA and its application to face recognition. Neurocomputing 71:191–196
17. Swets DL, Weng J (1996) Using discriminative eigenfeatures for image retrieval. IEEE T PAMI 18(8):831–836
18. Wu X-J, Kittler J, Yang J-Y, Messerm K, Wang S (2004) A new direct LDA (D-LDA) algorithm for feature extraction in face recognition. Proc Int Conf Pattern Recognit 4:545–548
19. Xiaogang W, Xiaoou T (2004) A unified framework for subspace face recognition. IEEE T PAMI 26(9):1222–1228
20. Ye J, Xiong T (2006) Computational and theoretical analysis of null space and orthogonal linear discriminant analysis. J Mach Learn Res 7:1183–1204
21. Zhao W, Chellappa R, Nandhakumar N (1998) Empirical performance analysis of linear discriminant classifiers. In: Proceedings of IEEE Conferences on Computer Vision and Pattern Recognition, pp 164–169
22. Sharma A, Paliwal KK (2008) A Gradient linear discriminant analysis for small sample sized problem. Neural Process Lett 27(1):17–24
23. Alsberg BK, Kvalheim OM (1994) Speed improvement of multivariate algorithms by the method of postponed basis matrix multiplication Part II. Three-mode principal component analysis. Chemom Intell Lab Syst 24:43–54
24. Carroll JD, Pruzansky S, Kruskal JB (1980) CANELINC: a general approach to multidimensional analysis of many-way arrays with linear constraints on parameters. Psychometrika 45:3–24
25. Giordani P, Kiers HAL (2004) Three-way component analysis of interval-valued data. J Chemom 18:253–264
26. Kiers HAL, Harshman RA (2009) An efficient algorithm for Parafac with uncorrelated mode-A components applied to large $I \times J \times K$ datasets with JK. J Chemom 23:442–447
27. Kiers HAL, Kroonenberg PM, Berge JMFT (1992) An efficient algorithm for Tuckals3 on data with large numbers of observation units. Psychometrika 57:415–422

28. Poon B, Amin MA, Yan H (2011) Performance evaluation and comparison of PCA Based human face recognition methods for distorted images. Int J Mach Learn Cybern 2(4):245–259
29. Sharma A, Paliwal KK (2012) A two-stage linear discriminant analysis for face-recognition. Pattern Recogn Lett 33(9):1157–1162
30. Pei D, Mi J-S (2011) Attribute reduction in decision formal context based on homomorphism. Int J Mach Learn Cybern 2(4): 289–293
31. Wang X, Dong L, Yan J (2012) Maximum ambiguity based sample selection in fuzzy decision tree induction. IEEE Trans Knowl Data Eng 24(8):1491–1505
32. Wang X, Dong C (2009) Improving generalization of fuzzy if-then rules by maximizing fuzzy entropy. IEEE Trans Fuzzy Syst 17(3):556–567
33. Sharma A, Imoto S, Miyano S (2012) A filter based feature selection algorithm using null space of covariance matrix for DNA microarray gene expression data. Curr Bioinform 7(3): 289–294
34. Sharma A, Imoto S, Miyano S, Sharma V (2011) Null space based feature selection method for gene expression data. Int J Mach Learn Cybern. doi:10.1007/s13042-011-0061-9