Chapter 6

Variable Selection and Feature Extraction Through Artificial Intelligence Techniques

Silvia Cateni, Marco Vannucci,
Marco Vannocci and Valentina Colla

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/53862

1. Introduction

The issue of variable selection has been widely investigated for different purposes, such as clustering, classification or function approximation becoming the focus of many research works where datasets can contain hundreds or thousands variables. The subset of the potential input variables can be defined through two different approaches: feature selection and feature extraction. Feature selection reduces dimensionality by selecting a subset of original input variables, while feature extraction performs a transformation of the original variables to generate other features which are more significant. When the considered data have a large number of features it is useful to reduce them in order to improve the data analysis. In extreme situations the number of variables can exceed the number of available samples causing the so-called problem of *curse of dimensionality* [1], which leads to a decrease in terms of accuracy of the considered learning algorithm when the number of features increases. The main reason for seeking for data reduction include the need to reduce calculation time of a given learning algorithm, to improve its accuracy [2] but also to deepen the knowledge of the considered problem, by discovering which factors actually affect it. A high number of contributions based on artificial intelligence, genetic algorithms, statistical approaches have been proposed in order to develop novel efficient variable selection methods that are suitable in many application areas. Section 1 and Section 2 provide a preliminary review of traditional and Artificial Intelligence–based feature extraction techniques and variable selection in order to demonstrate that Artificial Intelligence are often capable to outperform the widely adopted traditional methods, due to their flexibility and to their possibility of self-adapting to the characteristics of the available dataset. Finally in Section 4 some concluding remarks are provided.

2. Feature extraction

Feature extraction is a process that transforms high dimensional data into a lower dimensional feature space through the application of some mapping. Brian Ripley [3] gives the following definition of the feature extraction problem:

> "*Feature extraction is generally used to mean the construction of linear combinations $\alpha^T x$ of continuous features which have good discriminatory power between classes*".

In Neural Network research, as well as in other disciplines included in the Artificial Intelligence area, an important problem is finding a suitable representation of multivariate data. Feature extraction is used in this context in order to reduce the complexity and to give a simpler representation of data representing each component in the feature space as a linear combination of the original input variables. If the extracted features are suitably selected, then it is possible to work with the relevant information from the input data using a reduced dataset. The most popular feature extraction technique is the Principal Component Analysis (PCA) but many alternatives in the last years are been proposed. In the following sub-paragraphs several feature extraction approaches are proposed.

2.1. Principal Component Analysis

The Principal Component Analysis (PCA) was introduced by Karl Pearson in 1901 [4]. PCA consists into an orthogonal transformation to convert samples belonging to correlated variables into samples of linearly uncorrelated features. The new features are called *principal components* and they are less or equal to the initial variables. If data are normally distributed, then the principal components are independent. PCA mathematically transforms data by referring them to a different coordinate system in order to obtain on the first coordinate the first greatest variance and so on for the other coordinates [5]. Figure 1 shows an example of PCA in 2D. The original coordinate system (x,y) is transformed into the feature space (x', y') in order to have the maximum variance in the x' direction.

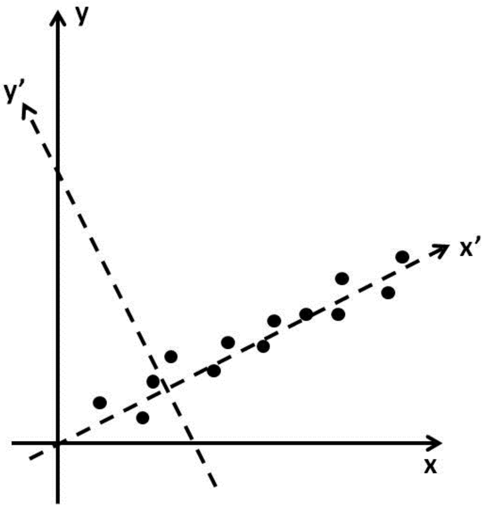

Figure 1. Example of PCA in 2D.

The main reason for the use of PCA concerns the fact that PCA is a simple non-parametric method used to extract the most relevant information from a set of redundant or noisy data. This method reduces the number of available variables by eliminating the last principal components that do not significantly contribute to the observed variability. Also, PCA is a linear transformation of data that minimizes the redundancy (which is measured through the covariance) and maximizes the information (which is measured through the variance). The principal components are new variables with the following properties:

1. each principal component is a linear combination of the original variables;
2. the principal components are uncorrelated to each other and also the redundant information is removed.

2.2. Linear Discriminant Analysis

While the PCA is unsupervised (i.e. it does not take into account class labels), the Linear Discriminant Analysis (LDA) is a popular supervised technique which is widely used in computer-vision, pattern recognition, machine learning and other related fields [6]. LDA performs an optimal projection by maximizing the distance between classes and minimizing the distance between samples within each class at the same time [7]. This approach reduces the dimensionality preserving as much of the class discriminatory information as possible. The main limitation of this approach lies in the fact that it can produce a limited number of feature projections (that is equal to the number of classes minus one). If more features are needed some other method should be employed. Moreover LDA is a parametric method and it fails if the discriminatory information lies not in the mean values but in the variance of data. When the dimensionality of data overcomes the number of samples, which is known as *singularity problem*, Linear Discriminant Analysis is not an appropriate method. In these cases the data dimensionality can be reduced by applying the PCA technique before LDA. This approach is called PCA+LDA [8, 9]. Other solutions dealing with the singularity problem include regularized LDA (RLDA) [10], null space LDA (NLDA) [11], orthogonal centroid method (OCM) [12], uncorrelated LDA (ULDA) [13].

2.3. Latent Semantic Analysis

Latent Semantic Analysis (LSA) was introduced by Deerwester et al. in 1990 [14] as a variant of the PCA concept. Firstly LSA was presented as a text analysis method when the features are represented by terms occurring in the considered text [2]. Subsequently LDA has been employed on image analysis [15], video data [16] and music or audio analysis [17]. The main objective of the LSA process is to produce a mapping into a "latent semantic space" also called *Latent Topic Space.* LSA finds co-occurrences of terms in documents to provide a mapping into the latent topic space where documents can be connected if they contain few terms in common respect to the original space. Recently Chen et al. [18] proposed a new method called Sparse Latent Semantic Analysis which selects only few relevant words for each topic giving a compact representation of topic-word relationships. The main advantage of this approach lies in the computational efficiency and in the low memory required for storing the projection matrix. In [18] the authors compare the Sparse Latent Semantic Analysis with LSA and LDA through experiments on different real world datasets. The obtained results demonstrate that Sparse LSA has similar performance with respect to LSA but it is more efficient in the projection computation, storage and it better explains the topic-world relashionships.

2.4. Independent Component Analysis

Independent Component Analysis (ICA) is an approach where the objective is to find a linear representation of non-gaussian data and the calculated components are statistically independent [19]. In literature at least three definitions of ICA has been given [20-22]:

i. General definition. ICA of the random vector consists of finding a linear transform **s=Wx** so that the components s_i are as independent as possible, in the sense of maximizing some functions $F(s_{i,} \dots s_n)$ that measures independence.
ii. Noisy ICA model. ICA of a random vector **x** consists of estimating the following generative model for the data **x=As+n** where the latent variables (components) s_i in the vector $s = (s_{1,} \dots, s_n)^T$ are assumed independent. The matrix **A** is a constant mxn mixing matrix, and **n** is a m-dimensional random noise vector.
iii. Noise-free ICA model. ICA of a random vector **x** consists of estimating the following generative model for the data: **x=As** where **A** and **s** are as in Definition 2.

The first definition is the most general one, as no a priori assumptions on the data are made. However it is an imprecise definition, as it is necessary to define a measure of independence for $s_{i.}$ The second definition reduces the ICA problem to an estimation of a latent variable method, but this estimate can be quite difficult; definition 3 is actually the most used one.

The possibility to identify a noise-free ICA approach is ensured by adding the following assumptions [22]:

1. All the independent components s_i must be non-gaussian (only one gaussian component should be accepted).
2. The number of observed mixtures must be greater or equal to the number of independent components.

ICA can be used to extract features finding independent directions in the input space. This objective is more difficult than using PCA approach, as in PCA the variance of data along a direction can be immediately calculated and it is maximised by PCA itself, while there is not straightforward metric for quantifying the independence of directions belonging to the input space [23]. Recently, in order to extract independent components, neural network algorithms have been adopted [24].

3. Variable selection

Variable selection approach reduces the dimension of a dataset of variables potentially relevant with respect to a given phenomenon by finding the best minimum subset without transform data into a new set. Variable selection points out all the inputs affecting the phenomenon under consideration and it is an important data pre-processing step in different fields such as machine learning [25-26], pattern recognition [27, 28], data mining [29], medical data [30] and many others. Variable Selection has been widely performed in applications such as function approximation [31], classification [32-34] and clustering [35]. The difficulty of extracting the most relevant variables is due mainly to the large dimension of the original variables set, the correlations between inputs which cause redundancy and finally the presence of variables which do not affect the considered phenomenon and thus, for instance in the case of the development of a model predicting the output of a give system, do not have any predictive power [36]. In order to select the optimal subset of input variables the following key considerations should be taken into account:

- **Relevance.** The number of selected variables must be checked in order to avoid the possibility to have too few variables which do not convey relevant information.
- **Computational efficiency.** If the number of selected input variables is too high, then the computational burden increases. This is evident when an artificial neural network is performed. Moreover including redundant and irrelevant variables the task of training an artificial neural network is more difficult because irrelevant variables add noise and slow down the training of the network.
- **Knowledge improvement.** The optimal selection of input variables contributes to a deeper understanding of the process behaviour.

To sum up, the optimal set of input variables will contain the fewest number of variables needed to describe the behaviour of the considered system or phenomenon with the minimum redundancy and with informative variables.

If the optimal set of input variables is identified, then a more accurate efficient, inexpensive and more easy interpretable model can be built.

In literature variable selection methods are classified into three categories: filter, wrapper and embedded methods.

3.1. Filter approach

Filter approach is a pre-processing phase which is independent of the learning algorithm that is adopted to tune and/or build the system (e.g. a predictive model) that exploits the selected variables as inputs. Filters are computationally convenient but they can be affected by overfitting problems. Figure 2 shows a generic scheme of the approach.

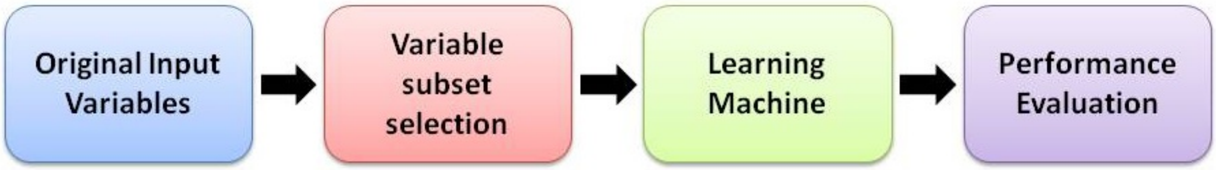

Figure 2. Generic scheme of filter methods.

The subset of relevant variables is extracted by evaluating the relation between input and output of the considered system. All input variables are classified on the basis of their pertinence to the target considering statistical tests [37, 38]. The main advantage of filter approach regards the low computational complexity ensuring speed to the model. On the other hand the main disadvantage of filter approach is that, being independent of the algorithm that is used to tune or build the model which is fed with the selected variables as inputs, this method cannot optimize the adopted model in the learning machine [39]. In the following subparagraphs some of the popular filter approaches presented in literature are described.

3.1.1. Chi-square approach

The chi-square approach [40] evaluates variables individually by measuring their chi-squared statistic. The test provides a score that follows a chi-square distribution with the objective to rank the set of input features. This approach is widely used but it does not take into account features interaction. If we assume that the class variable is binary the chi-squared value for scoring the belonging of variable *v* to the class *k* is evaluated as follows:

$$X^2(D,k,v) = \sum_{i=1}^{N}[\frac{(n_{i+}-\mu_{i+})^2}{\mu_{i+}} + \frac{(n_{i-}-\mu_{i-})^2}{\mu_{i-}}] \tag{1}$$

where *D* is the considered dataset, *N* is the number of the input variables, n_{i+} is the number of samples that have positive class for the variable *i* and finally μ_{i+} represents the expected value if there are any relationship between *v* and *k*.

In statistic the chi-squared test is used to verify if two events are independent. In feature selection chi-squared statistic performs a hypothesis test on the distribution of the class, as it relates to the measure of the variable under consideration; the null hypothesis represents an absence of correlation.

3.1.2. Correlation method

The correlation approach, used in feature selection, consists in calculating the correlation coefficient between the features and the target (or the class in the case of classification problems). A feature is selected if it is highly correlated with the class but not correlated with the remaining features [44]. There are two different approaches which evaluate the correlation between two variables: the classical linear correlation and the correlation based on information theory. Regard to the linear correlation coefficient, it is calculated by following equation:

$$c = \frac{\sum_i(x_i-\mu_{xi})(y_i-\mu_{yi})}{\sqrt{\sum_i(x_i-\mu_{xi})^2}\sqrt{\sum_i(y_i-\mu_{yi})^2}} \tag{2}$$

where *x*, *y* are the two considered variables, while μ_x and μ_y are their mean values. The linear correlation coefficient *c* lies in the range [-1, 1]. If the two variables are linearly correlated then |*c*|=*1*, while if they are independent *c* assumes a null value. This approach has two main advantages: it removes features having a very low correlation coefficient and it reduces redundancy. On the other hand, the linear correlation approach does not adequately outline non linear correlations, which often occur when treating with real world datasets.

3.1.3. *Information Gain*

Information Gain (IG) is widely used on high dimensional data, such as text classification [41]. It calculates the amount of information in bits concerning the class prediction when the only information available is the presence of a variable and the corresponding target (or class) distribution [42]. Also, it measures the expected decrease in entropy in order to decide how important a given feature is. An entropy function increases when the class distribution becomes more sparse and it can be recursively applied to find the subsets entropy. The following equation provides an entropy function which satisfies the two requirements.

$$H(D) = -\sum_{i=1}^{C} \frac{n_i}{n} \log({}^{n_i}/_{n}) \tag{3}$$

where D is the dataset, n is the number of instances included in D, n_i represents the members in class i and C is the number of classes. Moreover the following equation represents the entropy of the subsets.

$$H(D|X) = \sum_j \left(\frac{|x_j|}{n}\right) H(D|x - x_j) \tag{4}$$

where $H(D|x{=}x_j)$ represents the entropy correlated to the subset of instances which assumes a value of x_j for the feature x. For example, when x provides a good description of the class, the value which is associated to that feature assumes a low value of entropy in its class distribution. Finally the Information Gain is defined as the reduction in entropy as follows:

$$\mathrm{IG(X)=H(D)\text{-}H(D|X)} \tag{5}$$

High value of the *IG* indicates that *X* is a significant feature for the considered phenomenon [43].

3.2. Wrapper approach

While filter methods select the subset of variables in a pre-processing phase independently from the machine learning method that is used to build the model that should be fed with the selected variables, wrapper approaches consider the machine learning as a black box in order to select subsets of variables on the basis of their predictive power. The wrapper approach was introduced by Kohavi and John in 1997 [45] and the basic idea is to use the prediction performance (or the classification accuracy) of a given learning machine to evaluate the effectiveness of the selected subset of features. A generic scheme concerning wrapper approach is shown in Figure 3. Wrapper method is computationally more expensive than filter approach and it could be seen as a brute force approach. On the other hand, considering the learning machine as a black box, wrapper methods are simple and universal. The exhaustive search becomes unaffordable if the number of variables is too large. In fact, if the dataset contains *k* variables, 2^k possible subsets need to be evaluated, i.e 2^k learning processes to run. The following sub paragraphs treat some wrapper strategies commonly used.

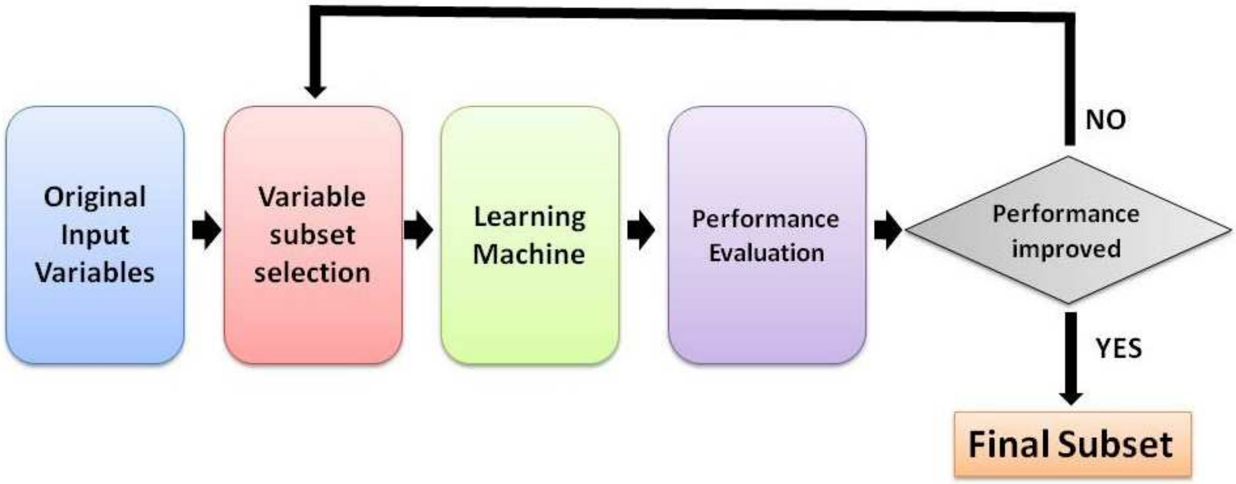

Figure 3. Generic diagram of wrapper approach.

3.2.1. Greedy search strategy

The Greedy search strategies can be divided into two different directions: Sequential Forward Selection (SFS) and Sequential Backward Selection (SBS). SFS approach starts with an empty set of features. The other variables are iteratively added into a larger subset until stopping criterion is reached. In general the adopted criterion is the improvement in accuracy. The proposed approach is computationally efficient and tests increasingly large sets in order to reach the optimal one. On the other hand SFS does not take into account all possible combinations but only selects the smallest subset: the risk arises to get trapped into a locally optimal point if the procedure prematurely ends [46]. SBS is the inverse of the forward selection approach. The process starts including all available features and then the less important variables are deleted one by one. In this case the importance of an input variable is determined by removing an input and evaluating the performance of the learning machine without it. If *k* is the number of the available input variables, the greedy search strategies needs, at maximum, $k(k+1)/2$ training procedures. When the SFS stops early it is less expensive than the SBS approach [47].

3.2.2. Genetic algorithm approach

Genetic algorithms (GAs) are efficient approaches for function minimization [43]. The genetic algorithm is a general adaptive optimization search technique and it is based on the Darwin Theory obtaining the optimal solution after iterative calculations. GAs create several populations of different possible solutions representing the so-called *chromosome* until an acceptable result is reached. A fitness function evaluates the goodness of a solution in evolution step. The crossover and mutation are operators that randomly affect the fitness score. In literature many wrapper approaches based on GA are proposed. Huang and Wang [48] present a genetic algorithm approach for feature selection and parameters optimization in order to improve the Support Vector Machine (SVM) classification accuracy [49]. Cateni et al. [50] present a method based on GAs that selects the best set of variables to be fed as input to a neural network. This approach is applied to a function approximation problem. The GA chromosomes are binary and their length corresponds to the number of available variables, also each gene is associated to an input. If the gene assumes unitary value it means that the corresponding input variable has been selected. The fitness function is represented by a feed-forward neural network [51] and the prediction performance is evaluated in terms of Normalized Square Root Mean Square Error (NSRMSE) [52]. The fitness function is computed for each chromosome of the population and crossover and mutation operators are applied. The crossover operator generates the son chromosome by randomly taking the genes values from the two parents, while mutation operation creates new individuals by randomly select a gene of the considered chromosome and switches it from 1 to 0 or vice-versa. The stop conditions include a fixed number of iterations or the achievement of a plateau for the fitness function. The generic scheme of the proposed approach is depicted in figure 4.

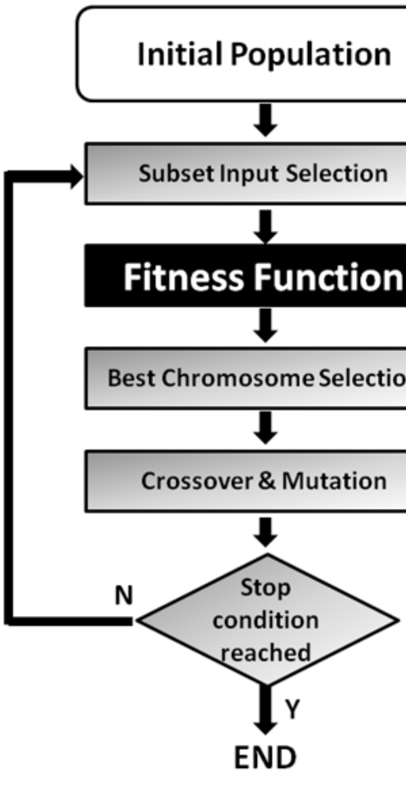

Figure 4. Genetic algorithm based approach.

The proposed approach has been tested on a synthetic database where three different targets (as non-linear combinations of variables) have been adopted. Moreover random noise, with gaussian distribution, has been added to each target variable in order to evaluate the effectiveness of the method. The obtained results demonstrate that the proposed approach selects all involved variables and the prediction error in terms of NSRMSE is about 4%. In [34] and [47] GAs are used not only for the selection of involved variables to be fed as inputs to the learning machine but also to optimize some important parameters of the learning algorithm used in a classification purpose. In particular in [34] a decision tree-based classifier [53] is adopted and the pruning level is optimized. Pruning [54] is used to increase the performance of the classifier by cutting unnecessary branches of the tree, by also improving the generalization capabilities of the decision tree. This approach has been tested on an industrial problem concerning the classification of the metal products quality on the basis on the product variables and process parameters. The results demonstrate the effectiveness of the proposed method obtaining a rate of misclassified products in the range 4%-6%. In [47] authors propose an automatic variable selection method which combines genetic algorithm and Labelled Self Organized Maps (LSOM) [55] for classification purpose. GAs are explored in order to find the best performing combination of variables in terms of accuracy concerning the classifier and for setting some important parameters of the SOM such as dimension of the net, topology function, distance function and others. The GA explores and computes the classification performance of different combinations of input features and Som Organized Map (SOM) parameters providing the optimal solution. The method has been tested on several databases belonging to the UCI repository [56]. The proposed approaches provide a satisfied classification accuracy given also comprehensions of the phenomenon under consideration by selecting the input variables which mainly affect the final classification.

3.3. Embedded approach

Unlike previous methods, embedded approach performs the variable selection in the learning machine. The variables are selected during the training phase, by thus reducing the computational cost and improving the efficiency during the phase of variables selection. The difference between embedded approach and wrapper approach is not always obvious but the main ones lies in the fact that embedded method requires iterative updates and the evolution of the model parameters are based on the performance of the considered model. Moreover wrapper approach considers only the model performance of the selected set of variables [57]. Figure 5 illustrates a generic scheme concerning the embedded approach.

As in embedded methods the learning machine and the variable selection should be incorporated the structure of the considered functions plays an important role [58]. For instance, in [59] the importance of a variable is measured through a bound that has a logic sense only for SVM-based classifiers. In [60] a novel neural network model is proposed called Multi-Layer Perceptrons using embedded feature selection (MLPs-EFS). Being an embedded approach, the feature selection part is incorporated into the training procedure.

With respect to the traditional MLPs this approach adds a pre-processing phase where each variable is multiplied by a scaling factor [61-62]. When the scaling factor is small then the features are considered redundant or irrelevant, while when it is large the features are relevant. Moreover another main advantage is that all optimization algoritms used for the MLPs are also suitable for MLPs-EFS. The authors demonstrate the effectiveness of the proposed approach compared to other existing methods such us Fisher Discriminant Ratio (FDR) associated to MLPs or SVM with Recursive Feature Elimination (RFE). Results demonstrate that MLPs-EFS outperform the other considered methods. Another good result of this approach lies in its generality, which allows to apply it to other type of neural networks.

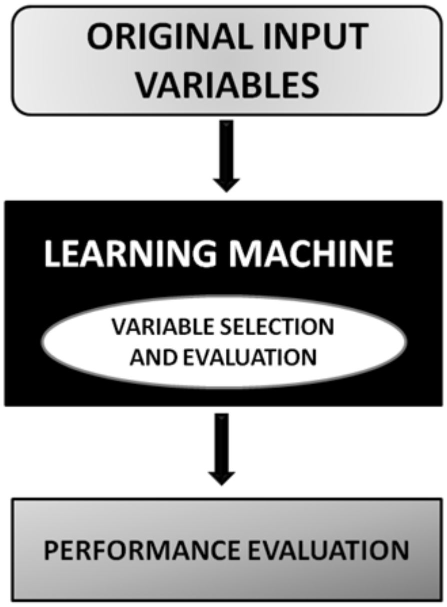

Figure 5. Generic scheme of embedded approach.

4. Conclusion

A survey about feature extraction and feature selection is proposed. The objective of both approaches concern the reduction of variables space in order to improve data analysis. This aspect becomes more important when real world datasets are considered, which can contain hundreds or thousands variables. The main difference between feature extraction and feature selection is that the first reduces dimensionality by computing a transformation of the original features to create other features that should be more significant, while feature selection performs the reduction by selecting a subset of variables without transforming them. Both traditional methods and their recent enhancements as well as some interesting applications concerning feature extraction and selection are presented and discussed. Feature selection improves the knowledge of the process under consideration, as it points out the variables that mostly affect the considered phenomenon. Moreover the computation time of the adopted learning machine and its accuracy need to be considered as they are crucial in machine learning and data mining applications.

Author details

Silvia Cateni, Marco Vannucci, Marco Vannocci and Valentina Colla
Scuola Superiore S.Anna,
TECIP - PERCRO Ghezzano, Pisa, Italy

5. References

[1] Bellman R. Adaptive Control Processes: A guided tour. Princeton University Press, 1961.

[2] Cunningham P. Dimension Reduction. Technical Report UCI-CSI-2007-7, August 8th, 2007. University College Dublin.

[3] Ripley B.D. Pattern Recognition and Neural Networks. Cambridge University Press, 1996.

[4] Pearson K. On lines and planes of closest fit to systems of points in space. Phisophical magazine 2, 1901, pp.559-572.

[5] Jolliffe I.T. Principal Component Analysis, Springer Series in Statistics, 2nd ed. Springer, NY 2002.

[6] Zhao J., Yu P. Shi, L., Li S. Separable linear discriminant analysis. Computational Statistics anda Data Analysis, 2012.

[7] Ye J., Ji S. Discriminant analysis for dimensionality reduction: an overview of recent developments. Biometrics Theory, Methods and Applications, Nov 2009.

[8] Belhumeour P.N., Hespanha J.P. Kriegman, D.J. Eigen faces vs Fisher faces: recognition using class specific linera projection. IEEE Transaction Pattern Analysis and Machine Intelligence, 19. pp.711-720. 1997.

[9] Wang X, Tang, X. A unified framework for subspace face recognition. IEEE Transactions on Pattern Analysis and machine Intelligence, 26. pp. 1222-1228, 2004.

[10] Guo Y., Hastie T., Tibshirani R. Regularized linear discriminant analysis and its application in microarrays. Biostatics, 8. pp.86-100, 2007.

[11] Chen L.F., Liao H.Y.M., Ko M.T., Lin J.C., Yu J.C. A new IDA-based face recognition system which can solve the small sample size problem. pattern Recognition, 33. pp.1713-1726, 2000.

[12] Park H., Jeon M., Rosen J.B. Lower Dimensional representation of text data based on centroids and least squares. BIT, 43. pp.1-22, 2003.

[13] Ye J. Characterization of a family of algorithms for generalized discriminant analysis on undersample problems. journal of Machine leraning Research, 6. pp.483-502, 2005.

[14] Deerwester S.C., Dumais S.T., Landauer T.K., Furnas G.W., Harshman A. Indexing by latent semantic analysis. Journal of the American society of Information Science, 41. pp.391-407. 1990.

[15] Heisterkamp D.R. Building a latent semantic index of an image database from patterns of relevance feedback. In ICPR (4), pp.134-137. 2002.

[16] Sahauria E., Zakhor A. Content Analysis of video using principal components. In ICIP, 3. pp. 541-545, 1998.

[17] Smaragdis P., Ray B., Shashanka M. A probabilistic latent variable model for acoustic Procesing at NIPS 2006, 2006.

[18] Chen X., Qi Y., Bai B., Lin Q., Carbonell J.G. Sparse Latent Semantic Analysis, SIAM, International Conference on Data Mining (SDM), 2011.

[19] Hyvannen A., Oja E. Independent Component Analysis: algorithms and Applications. Neural Networks, 13, pp.411-430, 2000.

[20] Hyvarinen A. Survey on Independent Component Analysis. Neural Computing Surveys 2, pp. 94-128, 1999.

[21] Comon P. Survey on independent Component Analysis: a new concept?, Signal Processing 36. pp.287-314, 1994.

[22] Jutten C., Herault J. Blind Separation of Sources, Part I: an adaptive algorithm ased on neuromimetic architecture, Signal processing, 24. pp. 1-10, 1991.

[23] Weingessel A., Natter M., Hornik, K. Using Independent Component Analysis for feature extraction and Multivariate data projection. Working Paper 16, Adaptive Information Systems and Modelling in economics and Management Science, August 1998.

[24] Karlhunen J. Neural approaches to independent component analysis and source separation. In 4th European Symposium on Neural Networks, pp. 249-266., Bruges, Belgium, 1996.

[25] Blum A.L., Langely P. Selection of relevant features and examples in machine learning. Artificial Intelligence, Vol. 69 pp-.245-271, 1977.

[26] John G.H., Kohavi R., Pfleger K. Irrelevant feature and the subset selection problem. Proc of 11th International conference on machine learning, pp.121-129, 1994.

[27] Bne-Bassant. Pattern recognition and reduction of dimensionality, Handbook of statistics II, Krisnaiah and Kanal eds, pp.773-791, 1982.

[28] Mitra P. Murthy C.A. and Pal, S.K. Unsupervised Feature selection using feature similarity, IEEE transaction on Pattern Analysis and machine intelligence. Vol. 24, pp.301-312, 2002.

[29] Dash M., Liu H. Feature selection for classification. Intelligent data analysis: an international journal. Vol. 11, N°3, pp.131-156, 1997.

[30] Puronnen S., Tsymbal A., Skrypnik, I. Advanced local feature selection in medical diagnostics. Proc. 13th IEEE Symposum computer-based medical diagnostics, pp.25-30, 2000.

[31] Sofge D.A., Elliot D.L. Improved Neural Modelling of real world Systems using genetic algorithm based Variable Selection, proc. Conference on Neural Networks and Brain, Oct 1998.

[32] Kwak N., Choi C.H. Input feature selection for classification problems. IEEE trans. on neural networks, Vol. 13, pp.143-159, 2002.

[33] Lin J.Y., Ke H.R., Chien B.C., Yang W.P. Classifier design with feature extraction using layered genetic programming. Expert System with Applications, 34. pp.1384-1393.

[34] Cateni S., Colla V., Vannucci M. Variable Selection througth Genetic Algorithms for classification purpose, IASTED International Conference on Artificial Intelligence and Applications, 2010, Innsbruck Austria, 15-17 February 2010.

[35] Wang S., Zhu J. Variable selection for model-based high dimensional clustering and its application on microarray data. Biometrics, 64. pp.440-448, June 2008.

[36] May R., Dandy G., Maier H. Review of input variable selection methods for artificial neural networks. Artificial Neural Network. Methodological Advances and Biomedical Applications. ISBN 978-953-307-243-2.

[37] Zhu Z., Ong Y.S., Dash M. Markov blanketembedded genetic algorithm for gene selection. Pattern Recognition, 40, pp.3236-3248, 2007.

[38] Khushaba R.N, Al-Ani A., Al-Jumaily A. Differential Evolution based Feature Subset Selection. 19th International Conference on Pattern Recognition, ICPR 2008. December 8-11, 2008. Tampa, Florida USA.

[39] Xiao Z., Dellandrea E., Dou W. Chen L. ESFS: A new embedded feature selection method based on SFS. Rapports de research, September 2008.

[40] Wu S., Flach P.A. Feature selection with labelled and unlabelled data. Proceeding of ECML7PKDD'02 Worshop on Integration and Collaboration Aspects of Data Mining, Decision Support and Meta-Learning, pp. 156-167, 2002.

[41] Yang, Y., Pedersen, J. O. A comparative study on feature selection in text categorization, Proc. of ICML'97, pages 412-420, 1997.

[42] Roobaert D., Karakoulas G, Nitesh V., Chawla V. Information Gain, Correlation and Support Vector Machines, StudFuzz 207, Springer- Verlag Berlin, pp. 463–470 (2006).

[43] Ladha L., Deepa T. Feature Selection methods and algorithms, International Journal on Computer Science and Engineering (IJCSE), Vol 3, N 5, May 2011.

[44] Yu L., Liu H. Feature Selection for high-dimensional data: a fast correlation-based filter solution. Proc. of the 20th International Conference on Machine Learning ICML-2003, Washington DC, 2003.

[45] Kohavi R., John G. Wrappers for feature selection. Artificial Intelligence, 97. pp. 273-324, December 1997.

[46] Guyon I., Elisseeff A. An introduction to variable and feature selection. The journal of Machine Learning Research, 3. pp.1157-1182, 2003.

[47] Cateni S., Colla V., Vannucci M. A Genetic Algorithm-based approach for selecting input variables and setting relevant network parameters of a SOM-based classifier. International Journal of Simulation Systems, Science & Technology. UKSim 4th European Modelling Symposium on Mathematical modelling and computer simulation, Vol.12, N°2, Aprile 2011.

[48] Huang C.L., Wang C.J. A GA-based feature selection and parameters optimization for support vector machines. Expert Systems with Applications 31. pp. 231-240, 2006.

[49] Vapnik V.N. The nature of statistical learning theory. New York, Springer, 1995.

[50] Cateni S. Colla V., Vannucci M. General purpose Input Variables Extraction: A Genetic Algorithm based Procedure GIVE A GAP, Proc of the 9th International Conference on Intelligence Systems design and Applications ISDA'09, November 30- December 2, 2009, Pisa, Italy.

[51] Patterson D. Artificial Neural Networks. Prentice Hall, New York Singapore 1996.

[52] Zhang Q., Benveniste A. Wavelet Networks. IEEE Transactions on Neural Network, Vol. 3, N°6, pp. 889-898, November 1992.

[53] Quinlan J.R. C4.5 Programs for Machine Learning. Morgan Kauffmann Publisher 1993.

[54] Kearns M.J., Mansur Y. A fast bottom up decision tree pruning algorithm with near-optimal generalization. In Proc. of the 15th International Conference on Machine Learning, 1998.

[55] Colla V., Vannucci M., Fera S., Valentini, R. Ca-treatment of AI killed steels: inclusion modification and application of Artificial Neural Networks for the prediction of clogging. Proc. 5th European Oxygen Steelmaking Conference EOSC'06, June 26-28, Aachen 2006, Germany, pp. 387-394.

[56] Asunction A., Newman D.J. UCI machine learning repository Irvine. CA: University of California, School of Information and Computer Science, 1997.
[http://archive.ics.uci.edu/ml7datasets.htm]

[57] Kwak N., Choi C. Input Feature Selection for Classification Problems. IEEE Transaction on Neural Networks, Vol. 13 N°1, January 2002.

[58] Navil Lal T., Chapelle O., Weston J., Elisseeff A. Embedded Methods. Feature Extraction, Foundations and Applications, Springer-Verlag, Berlin/Heidelberg Germany, 2006, pp. 137-165.

[59] Weston J., Mukherjee S. Chapelle O., Pontil M., Poggio T., Vapnik V. Feature Selection for SVMs. In S.A. Solla, T.K.Leen and K-R Muller editors. Advances in Neural Information Processing Systems, Volume 12, pp. 526-532, Cambridge,MA, USA, 2000, MIT Press.

[60] Bo L., Wang L., Jiao L. Multi-layer Perceptrons with embedded feature selection with application in Cancer Classification. Chinese Journal of Electronics, 15. 2006. pp. 832-835.

[61] Tibshirani R. Regression shrinkage and selection via the LASSO. Journal of the Royal Statistical Society (B), vol. 58, pp.267-288, 1996.

[62] Donoho D., Elad M. Optimally sparse representations in general (nonorthogonal) dictionaries by l1 minimization. Proceedings of the National Academic of Science, Vol.100, pp. 2197-2202, 2003.