

# Feature Selection with Kohonen Self Organizing Classification Algorithm

Francesco Maiorana

**Abstract**—In this paper a one-dimension Self Organizing Map algorithm (SOM) to perform feature selection is presented. The algorithm is based on a first classification of the input dataset on a similarity space. From this classification for each class a set of positive and negative features is computed. This set of features is selected as result of the procedure. The procedure is evaluated on an in-house dataset from a Knowledge Discovery from Text (KDT) application and on a set of publicly available datasets used in international feature selection competitions. These datasets come from KDT applications, drug discovery as well as other applications. The knowledge of the correct classification available for the training and validation datasets is used to optimize the parameters for positive and negative feature extractions. The process becomes feasible for large and sparse datasets, as the ones obtained in KDT applications, by using both compression techniques to store the similarity matrix and speed up techniques of the Kohonen algorithm that take advantage of the sparsity of the input matrix. These improvements make it feasible, by using the grid, the application of the methodology to massive datasets.

**Keywords**—Clustering algorithm, Data mining, Feature selection, Grid, Kohonen Self Organizing Map.

## I. INTRODUCTION

FEATURE selection techniques has become a real prerequisite to the classification, clustering or model building phase, typical of data mining applications, due to the increasing number of features present in many datasets coming from a wide range of applications. This is especially true in applications that have a high dimensional nature, such as sequence analysis, microarray analysis, spectral analysis, proteomics applications and KDT ones.

Some applications, such as KDT, are characterized by a very high number of features, which usually are the words of the documents. Each document is represented as a bag of words. The dataset could also have a lot of samples i.e. documents that must be clustered or classified. In this kind of problems reducing the number of features is of great importance since the presence of irrelevant features could produce noise, making the classification result prone to errors. Reducing the dimensionality of the dataset has also the advantage to speed up the mining process.

In other applications, such as mass spectrometry in

proteomics, the dataset is composed of thousand features that represent the mass/charge ratio, but few samples represented by the patients. The result is the so called high dimensionality small sample problem. This kind of problem suffers from the curse of dimensionality [1]. This term is used in literature to describe the fact that the number of samples needed to accurately describe a classification problem increases exponentially with the number of dimensions. Having a low number of samples such as the case of proteomics could lead to the discovery of a discriminative pattern between different populations even when the two populations are not statistically distinct. The same kind of problem appears in microarray experiments where a great amount of gene expression levels are present for few patients. Features selection is important for many reasons such as [2]:

- 1) Using all features to build up the model does not necessarily give the best performance. There is a break point in the number of features selected after which adding more features leads to worse performances since the added features are uninformative and they can cancel information in relevant features. This is the so called peaking phenomenon [3].
- 2) Some classification method requires a number of objects larger or equal to the number of features.
- 3) A feature selection step could avoid overfitting and improve model performance (prediction performance in case of supervised classification or better cluster detection in case of clustering)
- 4) A model with less features is faster to construct
- 5) A model with less variables is easier to interpret. This is especially important in bioinformatics application where a domain expert should interpret and validate the model.

This work will present a feature selection algorithm based on the SOM algorithm. The idea is to cluster the elements in the similarity space, then to use this result to extract, from each class a set of positive and negative features to be used to further cluster the elements in the reduced feature space.

This paper is organized as follows: section II presents a brief overview of feature selection techniques; section III outlines the implementation of the Kohonen SOM algorithm taken into account in the paper; section IV describes the proposed feature selection algorithm and presents some results on different datasets; section V draws the conclusions and future work.

Author is with the University of Catania, Dipartimento di Ingegneria Informatica e Telecomunicazioni, Viale A. Doria 6, Catania, 95125 Italy (phone: +390957382371; fax: +39 0957382397; e-mail: fmaioran@diit.unict.it).

## II. REVIEWS OF FEATURE SELECTION ALGORITHMS

Feature selection techniques are based on a selection of variables and do not alter their original space. In contrast, feature extraction techniques look for a mapping either linear or non linear of the original feature space into a projected space usually with lower dimension and more effective in describing the features. This mapping can be based on projection (such as Principal Component analysis) or compression (e.g. using information theory). The better known linear method is the Principal Component Analysis [4]. A review of non linear methods for feature extraction can be found in [5].

Feature selection techniques by just selecting a subset of variables do not alter the semantics of the variables allowing for an easy interpretation of the model carried out by a domain expert.

The feature selection techniques can be applied to supervised and unsupervised learning algorithms.

As reviewed in [2], [6] the feature selection techniques are usually grouped in four categories:

- 1) Filter techniques [7], [8], which select the relevant features by looking only at the intrinsic property of data. The classification algorithm is performed only on the selected features. These techniques can easily apply to very high dimensional datasets and are independent of the classification algorithm. The disadvantage is that the method ignores the interaction with the classifier and that most proposed techniques are univariate thus ignoring feature dependencies. Some examples are the information gain method [7] or a method based on a threshold of misclassification [8]
- 2) Wrapper methods [9], which embed the model hypothesis search within the feature subset search. Variable selection is performed in concert with the classification algorithm. The classification method is used to test the relevance of a variable. The variables that lead to the best performance are retained. These methods have the ability to take into account feature dependencies. The disadvantages of these techniques are the computational cost and the higher risk of overfitting than filter techniques.
- 3) Embedded techniques [10] in which the search for an optimal subset of features is built into the classifier construction, for example in the classification tree. These techniques, like the wrapper methods, are specific of a given learning algorithm.
- 4) Variable selection after classification. In this case model information or classification rules are used to find the most informative variables. Classification methods such as Support Vector Machine (SVM) or Discriminant Analysis (DA) carry on important information about the variables in the form of weights and regression coefficients.

A comprehensive survey of the feature selection algorithms is given in [11].

The paper will present a feature selection algorithm which broadly can be framed in the fourth category.

## III. BRIEF REVIEW OF THE SOM ALGORITHM

Kohonen SOM [12, 13, 14] are often used to cluster datasets in an unsupervised manner. This paper deals with on-line SOM since the batch version has some disadvantages such as the fact that it often represents an approximation of the on-line algorithm [15].

In the on-line version the weights are updated after the presentation of each input vector. In order to do this, the distance (usually the Euclidean one) is computed between the input vector and each weight vector as in (1).

$$d_k(t) = \|x(t) - w_k(t)\| \quad K = 1 \dots no \quad (1)$$

where no is the number of output neurons.

In the second step the algorithm searches for the winning neuron,  $d_w$ , i.e. the neuron that best matches the input neuron and is characterized by the minimum distance from the input vector.

$$d_w(t) = \min_k(d_k(t)) \quad K = 1 \dots no \quad (2)$$

In the third phase the algorithm updates the weights of the winning neuron and of the neurons that lie in a user defined neighborhood as follows:

$$w_k(t+1) = w_k(t) + \alpha(t)h_{kw}(t)\|x(t) - w(t)\| \quad K = 1 \dots no \quad (3)$$

where  $\alpha(t)$  is the learning rate that modulates the weight update, and  $h_{kw}$  is the neighborhood function that depends, given a time  $t$ , on the winning neuron  $w$  and the neuron under consideration  $k$ . Usually the output neurons are arranged in a bi-dimensional array; however some implementations have been proposed, e.g. [16, 17] which adopt a different topology of the network where the output neurons are arranged along a single layer (SL configuration).

Let us note that in the SL configuration the classes are given by the output neurons. This means that if, at the final cycle, the winning neuron mostly activated by the  $i^{\text{th}}$  item is the  $j^{\text{th}}$  neuron, then the input object belongs to class  $j$ .

In this scenery there is no topological similarity between output neurons since adjacent output neurons do not represent necessarily similar classes.

In the SL configuration the updating formula (3) is replaced by a neighborhood function that chooses the winning neurons and the ones (usually two or three) that are mostly activated by the current input object. The neighborhood function is not a topological but a logical one that finds the output neurons that are more close to the input vector. As neighborhood function the following one has been proposed in [16]:

$$h_{kw}(t) = \frac{1}{ord(k)^2} \quad (4)$$

Where  $ord(k)$  is the rank of weight vector  $k$  in the ordered vector of distance computed with formula 1).

In [16, 17] an automatic strategy to find the optimal number of classes is also proposed.

Fig. 1 shows a SL architecture with three input neurons and two output ones.

In [16, 17] the SL clustering algorithms work on both the feature and the similarity space. If the similarity space is used the algorithm allows us to perform a final step in which, for

each class, a set of features characterizing the elements belonging to the class can be found.

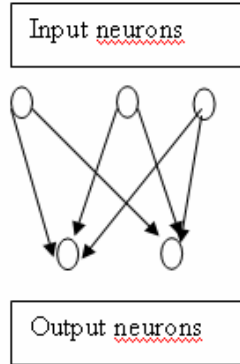


Fig. 1 A single layer network

In particular the algorithm performs the following steps:

For  $c = 1$  to the number of classes

For  $i = 1$  to the number of features

Compute:

$$c_i = \sum_{j=1}^{N_c} f_{j,i} \geq \text{threshold}_f. \quad (5)$$

$$d_i = \sum_{j=1}^{N_{nc}} f_{j,i} \geq \text{threshold}_f. \quad (6)$$

$$h_i = \sum_{j=1}^{N_c} f_{j,i} < \text{threshold}_{f_n}. \quad (7)$$

$$f_p = \frac{c_i}{\|N_c\|}. \quad (8)$$

$$f_{oc} = \frac{d_i}{\|N_{nc}\|}. \quad (9)$$

$$f_n = \frac{h_i}{\|N_c\|}. \quad (10)$$

where  $N_c$  is the set of documents belonging to class  $c$  and  $N_{nc}$  is the set of documents not belonging to class  $c$ , whereas  $f_{j,i}$  is the value of the feature  $i$  for document  $j$ . If the above inequality holds, the result is one, otherwise is zero.

The first summation counts the number of times feature  $i$  is greater than the  $\text{threshold}_f$  over all documents in the class, while the second summation counts over the documents that do not belong to the class. The third summation counts the number of documents in the class having feature  $i$  less than a  $\text{threshold}_{f_n}$ . For binary input matrix the two thresholds are the same so  $h_i$  is meaningful only for non-binary input matrices.

$f_p$  represents the fraction of documents in the class that have feature  $i$  greater than the  $\text{threshold}_f$ ;  $f_{oc}$  the fraction of documents not in the class that have feature  $i$  greater than the  $\text{threshold}_f$ ;  $f_n$  the fraction of documents in the class that have feature  $i$  less than the  $\text{threshold}_{f_n}$ .

A feature is labelled as positive if  $f_p$  is greater than a given

$\text{threshold}_p$ . For problems dealing with two classes a feature is labelled negative if  $f_n$  is greater than a given  $\text{threshold}_n$ . Usually  $\text{threshold}_n$  is greater than  $\text{threshold}_p$ . For clusterization with more than two classes a feature is labelled as negative if  $f_p - f_{oc} \leq \text{threshold}_{mc}$  and  $f_p < \text{threshold}_{\maxPos}$ .

#### IV. THE PROPOSED FEATURE SELECTION ALGORITHM

The first part of this section presents the idea of feature selection highlighting the key parameters that can be fine tuned to obtain the best results. The second part of the section presents some results obtained using both an in-house dataset used in a KDT application and public available datasets used in international competitions on feature selection

##### A. The Methodology

The idea for feature selection is to apply the proposed unsupervised clustering algorithm into two different spaces.

The first classification is performed over the similarity space. In doing this a similarity measure is chosen. By applying the selected similarity measure, a similarity matrix is computed containing the similarity of each sample against all the others. This similarity matrix is used as input for the unsupervised Kohonen algorithm. In this paper it is chosen to select the positive features arising from the similarity space classification that not necessarily are the ones from the feature space classification.

The results of the classification algorithms are used to extract, for each class, a set of positive and negative features. This will be hopefully composed of the most representative features, i.e. the set of features that characterize the majority of the elements of the class, or the set of features not present in the majority of the elements in the class or the features present in all other elements.  $\text{Threshold}_p$  is used to set the percentage of elements in the class that must have the common features. The value of this threshold will be optimized as explained below. The sets of positive and negative features are obtained using the classification results of the SL clustering algorithm on the similarity space. The collection of these features for all the classes is the result of the feature selection steps. In the original matrix all the features that do not belong to the selected set of representative features are removed. A second classification is performed, this time on the reduced feature space. The clustering results will be the final ones.

If the class membership of the element is known, at least for a subset of the examples, let say for a training set, one can use this information in a supervised manner to fine tune the threshold that affects the number of features extracted in the first phase of the algorithm. A modification of the threshold has an impact on the number of features extracted in the first phase and hence on the classification result on the reduced feature space. In fact, lowering the threshold has the effect of gathering more features; in some situation having more features produces a better classification result, in other one the correct classification rate decreases. By comparing the classification result with the reference one it is possible to fine tune the thresholds and hence find the best trade-off between correct classification rate and low number of features. An

extension could be to find the optimal threshold based on Receiver Operating Characteristic (ROC) [18] that takes into account specificity too.

In the experiment performed by lowering the value of threshold<sub>p</sub>, a greater number of features and a better classification result is obtained. There is a limit in the value of this threshold after which a future reduction implies an increase in the number of features, but a stable or even worse classification result.

It is also possible to use the knowledge of the class membership of a set of elements to choose the best similarity measure that affects the clustering result of the first phase and hence the feature selection step.

Various possibilities can be used for similarity measurement. In this paper it has been adopted a similarity in a broad sense defined by the sum of the minimum of each pair of vector components. The similarity matrix obtained is normalized between zero and one. In order to obtain a strict similarity measure we should normalize each row in such a way that the sum of its elements equals one. Other similarity measurements could also be considered such as the one used in collaborative filtering [19]. A future study will deeper investigate this aspect.

It is also for future study to investigate the impact of different classification algorithm (such as 2 or N dimensional SOM or other clustering algorithms) on the proposed feature selection methodology.

### B. Case Study and Results of the Proposed Feature Selection Algorithm

The presented algorithm has been applied to a problem in KDT. As reported in [20] a novel method to discover and evaluate hopefully new relationships from MEDLINE abstracts has been proposed.

In this scenario the proposed algorithm has been applied on a dataset of 3,528 rows and 262 columns. In this dataset the rows represent the abstracts to cluster and the columns represent a group of selected genes used as features to cluster the abstracts. The dataset was clustered in eighty classes. From this vector space model representation the similarity space representation was built. Various similarity measures have been considered. The best results were obtained with the sum of the minimum of the vector components. The similarity matrix used for classifications has a dimension of 3,528 X 3,528. In this matrix the total number of ones is 1,900,992 out of 12,446,784 elements equal to 15.27% of ones. The weighted average number of ones for each column (row) is 539 elements. The positive and negative features extracted were 142 out of 262. The cluster results in the feature space and in the reduced feature space are compared. The classification difference is 19%.

To further investigate the relevance of the method it was applied to the following datasets publicly available from previous international competitions on feature selection:

- 1) Dexter datasets [21]. This is a dataset for a two-class classification problem. The task is to filter texts about "corporate acquisitions" hence in the text domain. The original dataset is a subset of the Reuters text categorization benchmark with 9,947 features. To this

10,053 probe features were added for a total of 20,000. The input matrix is sparse integer and contains, for each document, the word indexes with the frequencies of the words in the documents. This dataset was one of the five datasets used in the NIPS 2003 feature selection challenge. The training and validation datasets are composed of 150 positive and 150 negative examples. The validation dataset contains 1,000 positive and 1,000 negative examples. The dataset is available at <http://www.nipsfsc.ecs.soton.ac.uk/datasets> and at the University of California Irvine Machine Learning Repository at <http://archive.ics.uci.edu/ml/datasets>.

- 2) Dorothea is a drug discovery dataset [21]. Chemical compounds represented by structural molecular features must be classified as active (binding to thrombin) or inactive. The dataset contains 100,000 integer features. This dataset was one of the five used in the NIPS 2003 feature selection challenge. The training dataset contains 78 positive examples and 722 negative ones. The validation dataset contains 34 positive examples and 316 negative ones. The test dataset contains 78 positive and 722 negative examples. The dataset is available at <http://www.nipsfsc.ecs.soton.ac.uk/dataset> and at the University of California Irvine Machine Learning Repository at <http://archive.ics.uci.edu/ml/datasets>.
- 3) Hiva was the most difficult dataset in the Model Selection workshop and performance prediction challenge inside the IEEE World Congress on Computational Intelligence (WCCI 2006) [22]. The dataset came from a drug discovery domain. It is a non-sparse matrix composed of 1,617 features. The training dataset contains 3,845 samples, the validation 384 ones and the test dataset 38,449 ones. The entries in the dataset are binary. A link is <http://www.modelselect.inf.ethz.ch/datasets.php>.

Since all the classification tasks are two classes problems,  $c_i$  and  $d_i$  exchange their role between the two classes, so only parameter  $f_p$  has been considered. The threshold used for all the above datasets in equation 1) and 2) was 0.9. The integer datasets were normalized between 0 and 1. In the first study the calculation of  $h_i$  in equation 3) has been avoided as the calculation of  $f_n$ .

Table I, II and III report the number of features selected and the classification performance on the reduced feature space compared with the available reference result, for different values of threshold<sub>p</sub>.

The training dataset is used, as described above, to choose the threshold<sub>p</sub> and hence the features. After the feature selection phase the same SL clustering algorithm is used to classify, in the reduced feature space, the training and validation dataset, and the classification results are compared with the classes indicated in the original dataset.

Dexter dataset contains 300 samples in the training and validation sets equally distributed in the two classes. The performance of the unsupervised cluster algorithm in the complete feature space was 0.99.

TABLE I  
NUMBER OF FEATURES SELECTED AND CLASSIFICATION PERFORMANCE  
ON DEXTER TRAIN AND VALIDATION DATASETS

Dexter Train		
Theshold <sub>p</sub>	#features	Performance
0.1	163/20,000	0.76
0.01	2,645	0.8033
0.008	3,879	0.8833
0.005	5,271	0.9567
0.004	7,751/20,000	0.99
Dexter validation		
Theshold <sub>p</sub>	# features	Performance
0.1	165/20,000	0.8967
0.01	3,299	0.89
0.008	4,181	0.8967
0.005	4,181	0.8967
0.004	7,847	0.5867

TABLE II  
NUMBER OF FEATURES SELECTED AND CLASSIFICATION PERFORMANCE ON  
DOROTHEA TRAIN AND VALIDATION DATASETS

Dorothea Train		
Theshold <sub>p</sub>	#features	Performance
0.1	1444/20000	0.95
0.05	5185	0.9537
0.04	7,655	0.9450
0.03	12,342	0.9387
0.02	22,781	0.8888
0.01	44,823	0.9712
Dorothea validation		
Theshold <sub>p</sub>	# features	Performance
0.1	2,511	0.9971
0.05	3,603	0.9971
0.04	4,866	0.9971
0.03	7,163	0.9971
0.02	15,780	0.9914
0.01	32,604	0.9886

TABLE III  
NUMBER OF FEATURES SELECTED AND CLASSIFICATION PERFORMANCE  
ON HIVA TRAIN AND VALIDATION DATASETS

Hiva Train		
Theshold <sub>p</sub>	#features	Performance
0.1	506/1617	0.6554
0.05	916	0.7277
0.04	1034	0.7350
0.03	1218	0.7415
0.02	1410	0.7467
0.01	1557	0.7493
Hiva validation		
Theshold <sub>p</sub>	# features	Performance
0.1	515/1617	0.6406
0.05	930	0.8203
0.04	1031	0.8333
0.03	1243	0.8307
0.02	1388	0.8516
0.01	1533	0.8516

Since the number of features is high, in order to increase the number of selected features the threshold was lowered up to consider a positive feature even if only one element in the class had it above the fixed threshold. With a threshold of 0.004 the algorithm identifies, in the training dataset, all the 7,751 training dataset features with at least one element different from zero reaching the correct classification rate of

0.99. For the validation dataset, with the same threshold, the algorithm correctly identifies all the 7,847 validation dataset features with at least one element different from zero. In this case however the algorithm obtains a poor 0.5867 correct classification rate, the same was obtained by considering the complete feature space. The best trade-off between number of features and classification performance, in both the training and validation set, is obtained with a threshold of 0.005 that identifies 5,271 features in the training dataset for a correct classification rate of 0.9567, and 4,181 features in the validation dataset for a correct classification rate of 0.8967. The number of features selected and the classification results are comparable to the ones presented in [23, 24].

Dorothea dataset contains 800 training samples and 350 validation ones. The features with at least one element different from zero are 88,119 in the training dataset and 77,113 in the validation dataset out of 100,000. The performance of the unsupervised clusterization algorithm is 0.995 in the training dataset and 0.9286 in the validation dataset. The best trade-off between the number of features selected and the classification performance is given by a threshold of 0.05 allowing us to select in the training dataset 5,185 features with a correct classification rate of 0.9537, and in the validation set 3,603 features with a correct classification rate of 0.9971. In the validation set the unsupervised algorithm, in the reduced feature space, outperforms the classification algorithm in the complete feature space. Moreover, the performance of the classification algorithm for the validation data set decreases if the threshold for positive features drops below 0.03.

Hiva dataset contains 3,845 training samples and 384 validation ones. The features with at least one element different from zero are 1,617 in the training dataset and 1,601 in the validation one out of 1,617. The performance of the unsupervised cluster algorithms is 0.7493 in the training dataset and 0.8516 in the validation one. The best trade-off between the number of feature selected and the classification performance is given by a threshold of 0.05 allowing us to select 916 features in the training dataset with a correct classification rate of 0.7277, and in the validation set to select 930 features for a correct classification rate of 0.8203.

Let us note that the knowledge of the classification results may be used to set the threshold parameters. This threshold could be applied on new data.

If the test set has a greater number of elements the threshold could be inferred by leaving the same proportion of elements in each class that must have the feature relevant. The proportion of the elements in each class can be hypothesized to be the same as the one in the training and validation set. It is for further study to investigate how much this hypothesis is relevant since in the proposed algorithm the features that are omitted for the elements belonging to one class, could be substituted by the features for the elements that do not belong to it (the parameter  $d_i$ ). In any case, when a new dataset is presented, a new unsupervised cluster analysis is built thus avoiding over-fitting problems. Moreover, the cluster results could be eventually used as a model to classify a new element: in this case the class for a new element will be the same class of the element most similar to the new one.

## V. CONCLUSIVE REMARKS AND FUTURE WORK

In this paper a feature selection method based on a clustering algorithm belonging to the Kohonen Self Organizing Feature map family has been proposed. The analysis was performed on various datasets of different nature. In all the datasets the proposed method obtained a good feature reduction with almost the same correct classification rate. This is an encouraging result to try to gain advantage by suitably reducing the feature space in problems characterized by input sparse matrix thus speeding up the mining process of massive datasets. Future works are: to extend the analysis to datasets with more samples; to further investigate the similarity measures to assess the impact on classification performance on the similarity space and on the feature selection step; to enrich the set of performance indexes to better evaluate the classification result; to extend the analysis to other datasets for multiclass classification problems; to extend the optimization technique by considering more parameters and other optimization techniques, such as genetic algorithms, in order to choose the optimum, and finally to study the impact of different classification algorithm on the proposed methodology. It is also planned to implement the proposed methodology in a grid infrastructure

## ACKNOWLEDGMENT

This work was supported by the program “ICT per l’Eccellenza dei territori - Intervento 1 – Piano ICT per l’Eccellenza del settore Hi-Tech nel territorio Catanese (ICT-E1)” promoted by the Italian Ministry of Innovation and by Catania Municipality.

## REFERENCES

- [1] T. Hastie, R. Tibshiranie, J. H. Friedman “The Elements of Statistical Learning. Data Mining, Inference and Prediction,” Springer, New York, 2003.
- [2] S. Smit, H. C. J. Hoefstloot, A. K. Smilde “Statistical data processing in clinical proteomics,” *Journal of Chromatography B*, Vol. 866, pp. 77-88, 2008.
- [3] A. Choudhary, M. Brun, J. Hua, J. Lowey, E. Suh, E. R. Dougherty “Genetic test bed for feature selection,” *Bioinformatics*, vol. 22, no. 7, pp 837–842, 2006.
- [4] K.V. Mardia, J. T. Kent, J. M. Bibby “Multivariate Analysis,” Academic Press, London, 1980.
- [5] L.J.P. Van der Maaten, E.O. Postma, H. J. van den Herik “Dimensionality reduction: a comparative review,” Submitted to *Neurocognition*, 2008.
- [6] Y. Saeys, I. Inza, P. Larranaga “A review of feature selection techniques in bioinformatics,” *Bioinformatics*, vol. 23 no. 19, pp. 2507–2517, 2007.
- [7] F. Model, P. Adorjàn, A. Olek, C. Piepenbrock, “Feature selection for DNA methylation based cancer classification,” *Bioinformatics*, vol. 17 (suppl. 1), pp. 157–164, 2001.
- [8] A. Ben-Dor, N. Friedman, Z. Yakhini “Class discovery in gene expression data” in *Proc of the 5th annual international conference on computational molecular biology*, pp 31–38, 2001.
- [9] R. Kohavi, G. H. John, “Wrappers for feature subset selection,” *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273–324, 1997.
- [10] I. Guyon, S.Gunn, M. Nikravesh, I. Zadeh, L. (Editors) “Feature Extraction, Foundations and Applications (Studies in Fuzziness and Soft Computing),” Chap. 6: Embedded methods. Springer, 2006.
- [11] I. Guyon, A. Elisseeff “An introduction to variable and feature selection,” *Journal of Machine Learning Research*, vol 3, pp. 1157-1182, 2003.
- [12] T. Kohonen “Self Organizing Maps,” Springer, 2000.
- [13] S. Kaski, J. Kangas, T. Kohonen “Bibliography of self organizing map (SOM) Papers: 1981 – 1997.” *Neural Computing Survey*, vol. 1, no. 3, pp. 102–350, 1998.
- [14] M. Oja, S. Kaski, T. Kohonen “Bibliography of self organizing map (SOM) papers: 1998 – 2001 Addendum,” *Neural Computing Survey*, vol. 3, no. 1, pp. 1–156, 2003.
- [15] M. Cottrel J.C. Fort, P. Letremy “Advantages and drawbacks of the batch Kohonen Algorithm,” in *Proc. 10th European Symp. On Artificial Neural Network*, pp. 223–230. Bruges (Belgium), 2005.
- [16] A. Faro, D. Giordano, F. Maiorana “Discovering complex regularities by adaptive Self Organizing classification,” *Proceedings of WASET*, vol. 4, pp. 27–30, 2005: <http://www.waset.org/pwaset/v4/v4-8.pdf>
- [17] A. Faro, D. Giordano, F. Maiorana “Discovering complex regularities from tree to semi – lattice classifications,” *International Journal of Computational Intelligence*, vol. 2, no. 1, pp. 34–39, 2005: <http://www.waset.org/ijci/v2-1-6.pdf>
- [18] T. Fawcett “An introduction to ROC analysis” *Pattern Recognition Letters* Vol. 27, pp. 861-874, 2006.
- [19] E. Spertus, M. Sahami, O. Buyukkocuten “Evaluating similarity measures: a large scale study in the Orkut Social Network,” In *Proc. of the eleventh ACM SIGKDD international conference on knowledge discovery in data mining*, pp. 678-684, 2005.
- [20] A. Faro, D. Giordano, F. Maiorana, C. Spanpinato, “Discovering Genes–Diseases Associations from Specialized Literature using the GRID.” To appear on *IEEE Transaction on Information Technology in Biomedicine*.
- [21] I. Guyon, “Design of experiments for the NIPS 2003 variable selection benchmark,” Technical Report, 2003. <http://www.nipsfsc.ecs.soton.ac.uk/papers/Datasets.pdf>.
- [22] I. Guyon, “Experimental design of the WCCI 2006 performance prediction challenge,” Technical Report, 2005.
- [23] I. Guyon, S. Gunn, A. Ben-Hur, G. Dror, G. “Result analysis of the NIPS 2003 feature selection challenge,” in *Proc NIPS*, 2004. [http://books.nips.cc/papers/files/nips17/NIPS2004\\_0194.pdf](http://books.nips.cc/papers/files/nips17/NIPS2004_0194.pdf).
- [24] I. Guyon, J. Li, T. Mader., P. A. Pletscher, G. Schneider, M. Uhr, “Competitive baseline methods set new standards for the NIPS 2003 feature selection benchmark,” *Pattern Recognition Letters*, vol 28, pp. 1438-1444, 2007.