Using a Clustering Similarity Measure for Feature Selection in High Dimensional Data Sets

Jorge M. Santos
ISEP - Instituto Superior de Engenharia do Porto
INEB - Instituto de Engenharia Biomédica, Porto, Portugal
LEMA - Laboratório de Engenharia Matemática, ISEP
Porto, Portugal
jms@isep.ipp.pt

Sandra Ramos
ISEP - Instituto Superior de Engenharia do Porto
LEMA - Laboratório de Engenharia Matemática, ISEP
Porto, Portugal
sfr@isep.ipp.pt

***Abstract*—Feature selection is a very important preprocessing step in data classification. By applying it we are able to reduce the dimensionality of the problem by removing redundant or irrelevant data. High dimensional data sets are becoming usual nowadays specially in bio-informatics, biology, signal processing or text classification, increasing the need for efficient feature selection methods. In this paper we study the applicability of a clustering validation measure, the Adjusted Rand Index (ARI), for this task comparing it with other methods based on statistical tests and on ROC curve. We have performed some experiments that show the validity of the proposed method.**

***Keywords*-feature selection; adjusted rand index; high dimensional data sets;**

I. Introduction

Feature selection or variable selection is the technique of selecting a subset of relevant feature for building robust learning. Feature selection is a very complex task. Several methods have been proposed to perform this task and to overcome the inherent difficulties in the classification of high dimensional data sets. These methods are usually divided in three groups: *filters*, *wrappers* and *embedded* ones. Filters are preprocessing steps, separated from the learning and classification process. They assign a score to each feature by computing the correlation or the mutual information between features or between features and the given labels. Single feature performance is also included in filters. In wrappers methods, like simulated annealing or genetic algorithms, features are grouped according to their contribution to the prediction performance of the learning machine. Wrappers are also separated from the learning and classification process. On the contrary, in embedded methods, the feature selection process is *embedded* in the learning process as part of the training phase. Decision trees are examples of embedded methods. A survey of all these methods can be found in [1].

The methods used in this work are included in the filters group since we apply them before the learning process. We will use a clustering validation measure, the Adjusted Rand Index, as a measure of correlation between each feature and the desired targets and compare it with feature selection performed with ROC curve and some statistical tests: Mann-Whitney-Wilcoxon (MWW), t-test and Kruskal-Wallis.

The Adjusted Rand Index (ARI) is a measure of agreement between partitions. Since the target data is partitioned by means of the labeling we can also use ARI to perform feature selection if we split each feature in non-overlapping equal intervals and compare the partition derived from the split with the one given by the targets. By doing this we are evaluating each feature's discriminant power and we can rank the features according to the computed ARI value. We can then select the most discriminant features to apply in our classification algorithm.

This work is organized as follows: the next section introduces the Adjusted Rand Index; Section 3 explains how we intend to use ARI for feature selection; Section 4 presents several experiments that show the applicability of the proposed measure when compared to the other methods with results detailed in Section 5. In the final section we draw some conclusions about the paper.

II. The Adjusted Rand Index

The Adjusted Rand Index is a performance index for cluster evaluation. There are several indices to perform this task. These indices are measures of correspondence between two partitions of the same data and are based on how pairs of objects are classified in a contingency table.

Let us consider a set of n objects $S = \{O_1, O_2, ..., O_n\}$ and suppose that $U = \{u_1, u_2, ..., u_R\}$ and $V = \{v_1, v_2, ..., v_C\}$ represent two different partitions of the objects in S such that $\cup_{i=1}^{R} u_i = S = \cup_{j=1}^{C} v_j$ and $u_i \cap u_{i'} = \emptyset = v_j \cap v_{j'}$ for $1 \leq i \neq i' \leq R$ and $1 \leq j \neq j' \leq C$. Given two partitions, U and V, with R and C subsets, respectively, the contingency Table I can be formed to indicate group overlap between U and V.

In Table I, a generic entry, t_{rc}, represents the number of objects that were classified in the rth subset of partition R and in the cth subset of partition C. From the total number

978-1-4244-8136-1/10/$26.00

Table I
THE CONTINGENCY TABLE FOR COMPARING PARTITIONS U AND V.

Partition		V				
	Group	v_1	v_2	$\cdots$	v_C	Total
U	u_1	t_{11}	t_{12}	$\cdots$	t_{1C}	$t_{1.}$
	u_2	t_{21}	t_{22}	$\cdots$	t_{2C}	$t_{2.}$
	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
	u_R	t_{R1}	t_{R2}	$\cdots$	t_{RC}	$t_{R.}$
Total		$t_{.1}$	$t_{.2}$	$\cdots$	$t_{.C}$	$t_{..} = n$

Table II
SIMPLIFIED 2×2 CONTINGENCY TABLE FOR COMPARING PARTITIONS U AND V.

Partition	V	
U	Pair in same group	Pair in different groups
Pair in same group	a	b
Pair in different groups	c	d

of possible combinations of pairs $\binom{n}{2}$ from a given set we can represent the results in four different types of pairs:
a - objects in a pair are placed in the same group in U and in the same group in V;
b - objects in a pair are placed in the same group in U and in different groups in V;
c - objects in a pair are placed in the same group in V and in different groups in U and;
d - objects in a pair are placed in different groups in U and in different groups in V.

This leads to an alternative representation of Table I as a 2×2 contingency table (Table II) based on a, b, c, and d.

The values of the four cells in Table II can be computed using the values of Table I by:

$$a = \sum_{r=1}^{R}\sum_{c=1}^{C}\binom{t_{rc}}{2} = \left(\sum_{r=1}^{R}\sum_{c=1}^{C} t_{rc}^2 - n\right)/2 \qquad (1)$$

$$b = \sum_{r=1}^{R}\binom{t_{r.}}{2} - a = \left(\sum_{r=1}^{R} t_{r.}^2 - \sum_{r=1}^{R}\sum_{c=1}^{C} t_{rc}^2\right)/2 \qquad (2)$$

$$c = \sum_{c=1}^{C}\binom{t_{.c}}{2} - a = \left(\sum_{c=1}^{C} t_{.c}^2 - \sum_{r=1}^{R}\sum_{c=1}^{C} t_{rc}^2\right)/2 \qquad (3)$$

$$\begin{aligned} d &= \binom{n}{2} - a - b - c = \binom{n}{2} - \sum_{r=1}^{R}\binom{t_{r.}}{2} - \sum_{c=1}^{C}\binom{t_{.c}}{2} + a \\ &= \left(\sum_{r=1}^{R}\sum_{c=1}^{C} t_{rc}^2 + n^2 - \sum_{r=1}^{R} t_{r.}^2 - \sum_{c=1}^{C} t_{.c}^2\right)/2 \end{aligned} \qquad (4)$$

where t_{rc} represents each element of the $R \times C$ matrix of Table I.

Using these four values we can calculate several different performance indices that we will present in the following paragraphs.

The Rand Index (RI), proposed by Rand [2], together with the well known Jaccard Index [3], were, and still are, popular indeces and probably the most used for cluster validation. We can easily compute the Rand Index between partitions U and V by:

$$RI_{(U,V)} = \frac{a+d}{a+b+c+d} \qquad (5)$$

and it basically weights those objects that were classified together and apart in both U and V. There are some well known problems with RI: the first as to do with the fact that the expected value of the RI of two random partitions does not take a constant value (say zero); the other is that the Rand statistic approaches its upper limit of unity as the number of clusters increases. To overcome these limitations some researchers have created several different measures. Examples are the Fowlkes-Mallows [4] Index $(a/\sqrt{(a+b)(a+c)})$ or the Adjusted Rand Index (ARI) proposed by Hubert and Arabie [5] as an improvement of RI. In fact ARI became one of the most successful cluster validation indices and in [6] it is recommended as the index of choice for measuring agreement between two partitions in clustering analysis with different numbers of clusters. We can compute the ARI index between partitions U and V, $ARI_{(U,V)}$, by

$$\frac{\binom{n}{2}(a+d) - [(a+b)(a+c)+(c+d)(b+d)]}{\binom{n}{2}^2 - [(a+b)(a+c)+(c+d)(b+d)]} \qquad (6)$$

or

$$\frac{\binom{n}{2}\sum_{r=1}^{R}\sum_{c=1}^{C}\binom{t_{rc}}{2} - \left[\sum_{r=1}^{R}\binom{t_{r.}}{2}\sum_{c=1}^{C}\binom{t_{.c}}{2}\right]}{\frac{1}{2}\binom{n}{2}\left[\sum_{r=1}^{R}\binom{t_{r.}}{2} + \sum_{c=1}^{C}\binom{t_{.c}}{2}\right] - \left[\sum_{r=1}^{R}\binom{t_{r.}}{2}\sum_{c=1}^{C}\binom{t_{.c}}{2}\right]} \qquad (7)$$

with expected value zero and maximum value 1.

III. USING ARI FOR FEATURE SELECTION

In classification problems the training data is partitioned by means of the given labels. We can also, following a criteria that we will explain later, make a partition for each feature and compare it with the partition given by the labels. Since ARI gives a measure of agreement between partitions we will use it to compare the partition given by the labels and the partition of each feature. By making these comparisons we can produce a rank of features.

We will start by explaining how to partition each feature. We will rank the feature values by splitting them in non-overlapping equal intervals (categories) that could be for

example as many as the number of classes. These intervals will define the partition to use, together with the class partition, in the computation of ARI index. Let us consider a simple example just to clarify this concept. Table III represents the values of two features (normalized in the interval $[0, 1]$) from a given data set with 12 elements with the respective class labels.

Table III
A SIMPLE EXAMPLE TO ILLUSTRATE THE USE OF ARI FOR FEATURE SELECTION

Element	a	b	c	d	e	f	g	h	i	j	k	l
Class label	1	1	1	1	2	2	2	2	3	3	3	3
feat1	0	0.3	0.1	0.5	0.2	0.4	0.7	0.5	0.9	1	0.7	0.4
feat2	1	0.8	0.9	0.7	0.2	0.4	0.4	0.5	0	0.1	0.1	0.2

We start by making partitions for both features and for the class labels. The partition defined by the class labels is $P_c = \{\{a, b, c, d\}, \{e, f, g, h\}, \{i, j, k, l\}\}$; the partitions for feature 1 will be $P_{feat1} = \{\{a, b, c, e\}, \{d, f, h, l\}, \{g, i, j, k\}\}$; and for feature 2 will be $P_{feat2} = \{\{e, i, j, k, l\}, \{f, g, h\}, \{a, b, c, d\}\}$. In this case we choose to split the feature values in 3 non-overlapping intervals but, as we will see later, we can choose different number of intervals. Using formulas 6 or 7 we then calculate the ARI values between each feature partition and P_c thus obtaining $ARI_{(P_c,P_{feat1})}$ and $ARI_{(P_c,P_{feat2})}$. We will then rank the features according to their ARI value. In the presented case $ARI_{(P_c,P_{feat2})} > ARI_{(P_c,P_{feat1})}$ therefore the feature with highest ARI is feat2 and so is the most discriminant feature.

ARI will give us the feature's discriminant power. Having ranked the existent features we select a certain number of the most discriminant ones to use in our classification algorithm. This approach is suitable for data sets with an extremely large number of features like those related with gene expression or text classification.

IV. EXPERIMENTS

As we mentioned earlier we will compare the selection of features performed by the application of ARI index and by some statistical tests. These univariate feature selection algorithms include feature ranking as principal selection mechanism because of its simplicity, easy implementation and good empirical success. These statistical methods estimate a score Δ_j for every distinct feature j on the data set and apply a selection rule based on the magnitude of Δ_j. An example of such a decision rule, as we stated earlier, is to rank the scores Δ_j from largest-to-smallest and select the top-ranked k features.

In the estimation of the Δ_j score we used the ARI and compare it with the t-test, the Mann-Whitney-Wilcoxon test [7] and the Receiver Operating Characteristic (ROC) curve procedure [8], for two-class problems and the Kruskall-Wallis test [7] for problems with more than two classes. For ROC curve procedure the Δ_j score is given by the area under the curve (AUC). The AUC is an important measure for the quality of separation. A high area under the ROC suggests good discriminative power of the model. In the cases of t, Mann-Whitney-Wilcoxon and Kruskall-Wallis tests the Δ_j was assessed by the p-value. Two independent sample t-test is a parametric procedure for comparing means of two independent populations. This test assumes that populations are normally distributed. If this is not true, the Central Limit Theorem can be used to justify that the sample sizes are large enough. Mann-Whitney-Wilcoxon test is a nonparametric test that does not depend on the Gaussian assumption for the populations and is used for determining whether there is a difference between two populations. The Kruskal-Wallis test, a simple extension of the Mann-Whitney-Wilcoxon test, is applied in cases of three or more populations. Note that the False Discovery Rate that controls the number of false positive features [9] was not applied here because we want to select a fixed number of features.

We applied these algorithms for feature selection in five data sets summarized in Table IV. Data sets Arcene, Dexter and Madelon can be found in the UCI repository [10]. Arcene is a Mass-spectrometric cancer data set, Dexter is a text classification problem and Madelon is an artificial data set. Data set Leukemia, a Microarray Gene Expression Data related with leukemia cancer, can be found in [11] and NCI60, also a cancer Microarray data, can be found in [12]. The data sets differ a lot among them specially in what concerns the number of features and the number of elements but the common characteristic is the large number of features.

Table IV
THE DATA SETS USED IN THE EXPERIMENTS.

Data set	number of elements	number of features	number of classes	number of elem. per class
Arcene	100	10000	2	44;56
Dexter	300	7751	2	150;150
Leukemia	72	7129	2	47;25
Madelon	2000	500	2	1000;1000
NCI60	64	6830	12	7;5;7;2;6;2;8;9;6;2;9;1

For the ARI and ROC feature selection methods we ranked the Δ_j scores from largest-to-smallest and select the top-ranked k features. For the other methods, being the Δ_j assessed by the p-value, we ranked the Δ_j scores from smallest-to-largest and selected also the top ranked k features. Actually we choose $k \in \{2, 5, 10, 20\}$ because we were interested in trying to obtain good results with a very small number of features. When applying ARI we performed several exploratory experiments to determine the ideal number of intervals (categories) to split each feature

and we find better results when choosing values for the number of intervals around the double of the number of classes. We can say also that the results are not significantly modified by different choices of the number of intervals. Therefore we choose 4 and 5 intervals for the two-class problems and 12, 24 and 30 intervals for the 12-class problem.

Since the purpose of this work is to compare the feature selection methods we thought it was better not to use different classification methods because we could incur in a unfair comparison by evaluating the results given by different methods and lose the main goal. For this reason we choose only neural networks (MLP's) as classification algorithms in all problems.

The architectures of the MLP's were the following: as many inputs as the number of features, one or two hidden layers and one output layer for the two-class problems and as many outputs as the number of classes for the multi-class problem. We performed experiments with different configuration: different number of hidden neurons in a given interval. This interval was chosen in order to assure not too complex network with acceptable generalization. For that purpose we used some criteria as guidelines and performed some preliminary experiments. As criteria we took into account the well-known rule of thumb $n_h = w/\epsilon$ (based on a formula given in [13]), where n_h is the number of hidden neurons, w the number of weights and ϵ the expected error rate. Other MLP characteristics were chosen following [14]: all neurons use the hyperbolic tangent as activation function; as risk functional we used the MSE and as learning algorithm the backpropagation of the errors. The inputs were all pre-processed in order to standardize them to zero mean and unit variance.

Several different configurations of the MLP's were used in the performed experiments. When using all the features of the data sets there is a need for more complex MLP's and therefore we choose to use two hidden layers with different number of neurons. Experiments with sets of selected features (2, 5, 10 and 20) were performed with a single layer with several different number of hidden neurons. The number of neurons for the experiments is shown on Table V.

In all experiments we used the 5-fold cross validation method. In this method in each run the data set is randomly split in five groups being each one, in five different stages, used for testing and the other four for training. Each experiment consisted of 20 runs of the algorithm. After the 20 runs the mean and standard deviation of the classification error were computed.

Table V
THE RANGE OF THE NUMBER OF NEURONS OF THE MLP'S USED IN THE EXPERIMENTS.

Data sets	Two hidden layers		One hidden layer	
	1st	2nd	2 and 5 features	10 and 20 features
Arcene	10-50	10-20	2-10	2-20
Dexter	10-80	10-40		
Leukemia	10-80	10-40		
Madelon	20-60	10-40		
NCI60	10-80	10-40		

V. RESULTS

In Table VI we show the best classification error mean and standard deviation (in brackets) of the performed experiments with the different feature selection methods with two-class and multi-class problems. We do not show, due to lack of space, the number of hidden neurons corresponding to these best results (in our opinion this is not an important subject of analysis in this work). However we must stress that the range of the number of neurons used in the experiments is wide enough to guaranty a fair comparison between the methods.

We present in the first row of results of Table VI the classification errors for the performed experiments using all the features.

The first point to notice is that the results clearly show that feature selection is a very important pre-processing step in high dimensional data sets. By performing feature selection and using a selected subset of features we were able to obtain better results than by using all features. This is mainly due to the fact that high dimensional data sets, with hundreds or thousands of features, contain high degree of irrelevant and redundant information which may degrade the performance of the neural network. This fact is more visible in data sets Leukemia and Madelon.

The second interesting point is that the feature selection based on ARI was able to achieve in overall terms similar results than those obtained by the other more common methods. Actually, in three data sets the results for ARI are the best ones.

Results also show that, with fewer exceptions, the classification errors becomes larger as we use less features. In fact the best results are all with 20 (one with 10) features. This could eventually mean that we have choose a very small number of features and probably we should make some experiments with more features to try to get even better results. However, the purpose of this work was to evaluate the applicability of ARI for feature selection and not to try to obtain the best classification results or even to beat other classification algorithms. This is the reasons for just having used MLP's in our experiments.

For data set NCI60 the best results were obtained with the Kruskal method (remember that this method is used only for multi-class problems.) The results for methods Mann-Whitney-Wilcoxon and *t*-test are very similar. The comparison between methods allows us to state, based on the performed experiments, that ROC presents the worst results

Table VI
CLASSIFICATION ERRORS (MEAN AND STANDARD DEVIATIONS OF THE 20 RUNS) FOR THE PERFORMED EXPERIMENTS. THE BEST RESULTS FOR EACH DATA SET ARE IN **BOLD**.

Method	Selected features	Data sets				
		Arcene	Dexter	Leukemia	Madelon	NCI
	All	18.20 (2.82)	14.17 (1.96)	5.56 (2.07)	43.96 (0.80)	51.56 (5.05)
ROC	2	24.30 (2.00)	30.77 (1.36)	6.25 (0.73)	38.08 (0.16)	
	5	25.80 (3.08)	19.37 (1.06)	7.36 (1.86)	33.34 (0.41)	
	10	27.40 (1.84)	15.77 (1.10)	8.61 (1.94)	35.30 (0.79)	
	20	23.10 (1.73)	15.87 (0.80)	6.39 (1.49)	38.37 (0.94)	
MWW	2	34.00 (4.00)	17.73 (0.78)	5.83 (0.59)	37.91 (0.22)	
	5	34.00 (3.40)	20.73 (1.20)	4.03 (0.69)	31.92 (0.66)	
	10	**13.60** (2.32)	13.07 (0.93)	4.03 (0.79)	17.02 (0.60)	
	20	14.40 (2.88)	12.03 (1.18)	3.89 (0.59)	18.91 (0.65)	
t-test	2	32.10 (2.02)	27.03 (0.48)	8.86 (0.90)	37.97 (0.28)	
	5	31.80 (2.44)	27.40 (0.52)	5.71 (1.35)	30.05 (0.56)	
	10	31.20 (2.25)	16.53 (1.08)	5.57 (1.71)	17.02 (0.60)	
	20	17.60 (2.88)	12.33 (1.05)	6.86 (1.62)	18.91 (0.65)	
Kruskal	2					69.53 (4.18)
	5					52.19 (5.90)
	10					50.47 (5.76)
	20					**40.16** (6.59)
ARI-4	2	24.90 (1.97)	25.87 (1.17)	1.86 (0.96)	38.02 (0.23)	
	5	21.30 (2.71)	17.50 (0.76)	1.71 (1.62)	31.48 (0.49)	
	10	22.20 (2.94)	13.50 (0.74)	4.57 (1.31)	**16.81** (0.33)	
	20	22.10 (2.23)	11.53 (1.25)	3.14 (0.90)	19.65 (0.73)	
ARI-5	2	26.10 (1.10)	25.53 (0.57)	7.57 (2.13)	37.92 (0.22)	
	5	25.60 (1.58)	20.60 (0.78)	4.43 (1.05)	31.92 (0.33)	
	10	20.30 (2.26)	13.93 (0.93)	3.43 (1.00)	17.33 (0.87)	
	20	20.50 (3.66)	**10.87** (1.12)	**1.57** (1.42)	19.69 (0.66)	
ARI-12	2					71.56 (5.60)
	5					65.78 (6.89)
	10					64.22 (7.64)
	20					53.59 (4.72)
ARI-24	2					73.75 (4.15)
	5					74.84 (6.36)
	10					62.81 (4.15)
	20					58.13 (6.33)
ARI-30	2					71.09 (3.55)
	5					77.19 (4.67)
	10					72.81 (5.99)
	20					62.34 (8.94)

and ARI the best ones.

Feature selection based on ARI also have the advantage that one can select the number of intervals (categories) in order to try to obtain even better results. This is a subject that we will explore in a future work.

Finally we must say that, despite the fact that our goal was not to achieve the best results on the classification problems, the results for Leukemia are better than those published in [11] and other works.

VI. CONCLUSIONS

The purpose of this work was to evaluate the suitability of ARI to perform feature selection.

We have presented the ARI index and explained how we can use it to perform feature selection by ranking the computed ARI for all the features with comparison to the labeling targets. The computed ARI will give us a measure of correlation between features and labels.

We have compared the results of the classification with neural networks using the features selected by ARI and other methods that also measure the correlation between the

observed data values.

Results show that ARI is a valid measure for feature selection having obtained better results when compared to other well known methods. We also obtained some good results using other classification algorithms but we do not present them here because the purpose of this work was only to compare the feature selection task with only one classification algorithm. We can also obtain better results by choosing a larger number of features. In overall terms we must say that the results are very promising.

Finally, we must say that we use this index in our daily experiments not only in feature selection but also as a measure of performance of the classification algorithms. We encourage all researchers to include ARI as a tool in their classification processes.

REFERENCES

[1] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[2] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, vol. 66, pp. 846–850, 1971.

[3] P. Jaccard, "Étude comparative de la distribution florale dans une portion des alpes et des jura," in *Bulletin del la Société Vaudoise des Sciences Naturelles*, 1901, no. 37, pp. 547–579.

[4] E. Fowlkes and C. Mallows, "A method for comparing two hierarchical clusterings," *Journal of the American Statistical Association*, vol. 78, pp. 553–569, 1983.

[5] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, no. 1, pp. 193–218, 1985.

[6] G. Milligan and M. Cooper, "A study of the comparability of external criteria for hierarchical cluster analysis," *Multivariate Behavioral Research*, vol. 21, pp. 441–458, 1986.

[7] D. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedure*, 2nd ed. Chapman & Hall, 2000.

[8] M. Zweig and G. Campbell, "Receiver operating characteristic (roc) plots: A fundamental evaluation tool in clinic medicine," *Clinical Chemistry*, vol. 39, no. 4, pp. 561–577, 1993.

[9] S. Dudoit and M. van der Laan, *Multiple Testing Procedure and Applications to Genomics*, ser. Springer Series in Statistics. Springer, New York, 2008.

[10] C. Blake, E. Keogh, and C. Merz, "UCI repository of machine learning databases," http://www.ics.uci.edu/˜mlearn/MLRepository.html, 1998.

[11] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, and E. Lander, "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, 1999.

[12] NCI60, "Stanford NCI60 cancer microarray project," http://genome-www.stanford.edu/nci60/, 2000.

[13] E. Baum and D. Haussler, "What size net gives valid generalization?" *Neural Computation*, vol. 1, no. 1, p. 151160, 1990.

[14] C. M. Bishop, *Neural Networks for Pattern Recognition*. N.Y.: Oxford University Press, 1996.