

A Comparative Study of Some Issues Concerning Algorithm Recommendation using Ranking Methods

Carlos Soares and Pavel Brazdil

LIACC/Faculty of Economics, University of Porto, R. Campo Alegre 823, 4150-180
Porto, Portugal, {csoares,pbrazdil}@liacc.up.pt

Abstract. Cross-validation (CV) is the most accurate method available for algorithm recommendation but it is rather slow. We show that information about the past performance of algorithms can be used for the same purpose with small loss in accuracy and significant savings in experimentation time. We use a meta-learning framework that combines a simple IBL algorithm with a ranking method. We show that results improve significantly by using a set of selected measures that represent data characteristics that permit to predict algorithm performance. Our results also indicate that the choice of ranking method has a smaller effect on the quality of recommendations. Finally, we present situations that illustrate the advantage of providing recommendation as a ranking of the candidate algorithms, rather than as the single algorithm which is expected to perform best.

1 Introduction

Ideally, we would like to be able to identify or design the single best algorithm to be used in all situations. However, both experimental results [1] and theoretical work [2] indicate that this is not possible. Therefore, the choice of which algorithm(s) to use depends on the data set at hand and systems that can provide such recommendations would be very useful. We could reduce the problem of algorithm recommendation to the problem of performance comparison by estimating the performance of all the algorithms on the data currently available, assuming that it is representative of future data. Cross-validation (CV) is the most accurate method available for that purpose. However, it is not usually feasible in practice because there are too many algorithms to try out, some of which may be quite slow. The problem is exacerbated in the iterative process of analyzing large amounts of data, as it is common in Knowledge Discovery in Databases.

Another approach to algorithm recommendation involves the use of meta-knowledge, that is, knowledge about the performance of algorithms. This knowledge can be either of theoretical or of experimental origin, or a mixture of both. The rules described by [3] for instance, captured the knowledge of experts concerning the applicability of certain classification algorithms. More often the meta-knowledge is of experimental origin, obtained by meta-learning on past performance information of the algorithms, i.e., performance of the algorithms on data previously analyzed [4, 5]. Its objective is to capture certain relationships between the measured data set characteristics and the performance of the algorithms. As was demonstrated, meta-knowledge can be used to give useful predictions with a certain degree of success.

In this paper we follow the meta-learning approach. We adopt a framework which uses the IBL algorithm as a meta-learner. The performance and the usefulness of meta-learning for algorithm recommendation depends on several issues. Here we investigate the following hypotheses:

- **data characterization:** can we improve performance by selecting and transforming features that we expect to be relevant?
- **ranking method:** given that there are several alternatives, can we single out one which is better than the others?
- **meta-learning:** are there advantages in using meta-learning, when compared to other alternatives?
- **type of recommendation:** is there advantage in providing recommendation in the form of ranking, rather than recommending a single algorithm?

We start by describing the data characteristics used (Section 2). In Section 3, we motivate the choice of recommending a ranking of the algorithms, rather than a single algorithm. We also describe the IBL ranking framework used and the ranking methods compared. Ranking evaluation is described in Section 4. Next, we describe the experimental setting and present results. In Section 6, we present some conclusions.

2 Data Characterization

The most important issue in meta-learning is probably data characterization. We need to extract measures from the data that characterize relative performance of the candidate algorithms, and can be computed significantly faster than running those algorithms. It is known that the performance of different algorithms is affected by different data characteristics. For instance, k-Nearest Neighbor will suffer if there are many irrelevant attributes [6].

Most work on meta-learning uses general, statistical and information theoretic (GSI) measures or *meta-attributes* [5, 7]. Examples of these three types of measures are number of attributes, mean skewness and class entropy, respectively [8]. Recently, other approaches to data characterization have been proposed (e.g. *landmarkers* [9]) which will not be considered here.

As will be described in the next section, we use the k-Nearest Neighbor algorithm for meta-learning, which, as mentioned above, is very sensitive to irrelevant and noisy attributes. Therefore, we have defined a small set of measures to be used as meta-features, using a knowledge engineering approach. Based on our expertise on the learning algorithms used and on the properties of data that affect their performance, we select and combine existing GSI measures to define *a priori* a small set of meta-features that are expected to provide information about those properties. The measures and the properties which they are expected to represent are:

- The *number of examples* discriminates algorithms according to how scalable they are with respect to this measure.
- The *proportion of symbolic algorithms* is indicative of the preference of the algorithm for symbolic or numeric attributes.
- The *proportion of missing values* discriminates algorithms according to how robust they are with respect to incomplete data.
- The *proportion of numeric attributes with outliers* discriminates algorithms according to how robust they are to outlying values, which are possibly due to noise¹. An attribute is considered to have outliers if the ratio of the variances of mean value and the α -trimmed mean is smaller than 0.7. We have used $\alpha = 0.05$.
- The *entropy of classes* combines information about the number of classes and their frequency, measuring one aspect of problem difficulty.
- The *average mutual information of class and attributes* indicates the amount of useful information contained in the symbolic attributes.
- The *canonical correlation of the most discriminating single linear combination of numeric attributes and the class distribution* indicates the amount of useful information contained in groups of numeric attributes.

More details about the basic features used here can be found in [8]. We note all three proportional features shown above represent new combinations of previously defined data characteristics.

¹ Note that we have no corresponding meta-attribute for symbolic attributes because none was available.

3 Meta-Learning Ranking Methods

Here we have used the meta-learning framework proposed in [10]. It consists of coupling an IBL (k-NN) algorithm with a ranking method. The adaptation of k-NN for ranking is simple. Like in the classification version, the distance function is used to select a subset of cases (i.e. data sets) which are most similar to the one at hand. The rankings of alternatives (i.e. algorithms) in those cases are aggregated to generate a ranking which is expected to be a good approximation of the ranking in the case at hand (i.e. is expected to reflect the performance of the algorithms on the data set at hand).

Several methods can be used to aggregate the rankings of the selected neighbors. [10] propose a ranking method specific for multicriteria ranking of learning algorithms. Here we will focus on three ranking methods that take only accuracy into account [11]. These methods represent three common approaches to the comparison of algorithms in Machine Learning, as described next.

Average Ranks Ranking Method This is a simple ranking method, inspired by Friedman’s M statistic [12]. For each data set we order the algorithms according to the measured error rates² and assign ranks accordingly. The best algorithm will be assigned rank 1, the runner-up, 2, and so on. Let r_j^i be the rank of algorithm j on data set i . We calculate the *average rank* for each algorithm $\bar{r}_j = (\sum_i r_j^i) / n$, where n is the number of data sets. The final ranking is obtained by ordering the average ranks and assigning ranks to the algorithms accordingly.

Success Rate Ratios Ranking Method As the name suggests this method employs ratios of success rates (or accuracies) between pairs of algorithms. For each algorithm j , we calculate $SRR_j = \sum_k \sqrt[n]{\prod_i SR_j^i / SR_k^i} / m$ where SR_j^i is the accuracy of algorithm j on data set i , n is the number of data sets and m is the number of algorithms. The ranking is derived directly from this measure, which is an estimate of the average advantage/disadvantage of algorithm j over the other algorithms. A parallel can be established between the ratios underlying this method and performance scatterplots that have been used in some empirical studies to compare pairs of algorithms [13]

Significant Wins Ranking Method This method builds a ranking on the basis of results of pairwise hypothesis tests concerning the performance of pairs of algorithms. We start by testing the significance of the differences in performance between each pair of algorithms. This is done for all data sets. In this study we have used paired t tests with a significance level of 5%. This is the highest of the most commonly used values for the significance level not only in AI, but in Statistics in general [12]. We have opted for this significance level because we wanted the test to be relatively sensitive to differences but, at the same time, as reliable as possible. We denote the fact that algorithm j is significantly better

² The measured error rate refers to the average of the error rates on all the folds of the cross-validation procedure.

than algorithm k on data set i as $SR_j^i \gg SR_k^i$. Then, we construct a *win table* for each of the data sets as follows. The value of each cell, $W_{j,k}^i$, indicates whether algorithm j wins over algorithm k on data set i at a given significance level and is determined in the following way:

$$W_{j,k}^i = \begin{cases} 1 & \text{iff } SR_j^i \gg SR_k^i \\ -1 & \text{iff } SR_k^i \gg SR_j^i \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Note that $W_{j,k}^i = -W_{k,j}^i$ by definition. Next, we calculate $pw_{j,k}$ for each pair of algorithms j and k , by dividing the number of data sets where algorithm j is significantly better than algorithm k by the number of data sets, n^3 . This value estimates the probability that algorithm j is significantly better than algorithm k . The ranking is obtained by ordering the $pw_j = (\sum_k pw_{j,k}) / (m - 1)$ obtained for each algorithm j , where m is the number of algorithms. The kind of tests underlying this method is often used in comparative studies of classification algorithms.

In Section 5 we present the results of an empirical study addressing the following hypotheses:

- Given the sensitivity of the Nearest-Neighbor algorithm to the quality of the attributes, the subset of meta-features selected is expected to provide better results than the complete set which is commonly used.
- The SRR ranking method is expected to perform better than the other two methods because it exploits quantitative information about the differences in performance of the algorithms.
- Our meta-learning approach is expected to provide useful recommendation to the users, in the sense that it enables them to save time without much loss in accuracy.

The results are obtained with the evaluation methods described in the next section.

4 Evaluation of Rankings and Ranking Methods

Ranking can be seen as an alternative ML task, similar to classification or regression, which must therefore have appropriate evaluation methods. Here we will use two of them. The first one is the methodology for evaluating and comparing ranking methods that has been proposed earlier for meta-learning [11]. The rankings recommended by the ranking methods are compared against the true observed rankings using Spearman's rank correlation coefficient [12]. An interesting property of this coefficient is that it is basically the sum of squared errors, which can be related to the commonly used error measure in regression.

³ A more formal definition is given by $pw_{j,k} = (\sum_i I\{W_{j,k}^i = 1\}) / n$ where I is the standard indicator function, which returns 1 if the condition given as argument is true and 0 otherwise.

Furthermore, the sum is normalized to yield more meaningful values: the value of 1 represents perfect agreement and -1, perfect disagreement. A correlation of 0 means that the rankings are not related, which would be the expected score of the random ranking method. We note that the performance of two or more algorithms may be different but not with statistical significance. To address this issue, we exploit the fact that in such situations the tied algorithms often swap positions in different folds of the N -fold cross-validation procedure which is used to estimate their performance. Therefore, we use N orderings to represent the true ideal ordering, instead of just one. The correlation between the recommended ranking and each of those orderings is calculated and its score is the corresponding average. To compare different ranking methods we use a combination of Friedman’s test and Dunn’s Multiple Comparison Procedure [12] that is applied to the correlation coefficients.

The second evaluation method is based on an idea which is quite common in Information Retrieval. It assumes that the user will select the top N alternatives recommended. In the case of ranking algorithms, the performance of the top N algorithms of a ranking will be the accuracy of the best algorithm in that set.

5 Results

Before empirically investigating the hypotheses in the beginning of this paper, we describe the experimental setting.

Our meta-data consists of 53 data sets mostly from the UCI repository [14] but including a few others from the METAL project⁴ (SwissLife’s Sisyphus data and a few applications provided by DaimlerChrysler). Ten algorithms were executed on those data sets⁵: two decision tree classifiers, C5.0 and Ltree, which is a decision tree that can introduce oblique decision surfaces; the IB1 instance-based and the naive Bayes classifiers from the MLC++ library; a local implementation of the multivariate linear discriminant; two neural networks from the SPSS Clementine package (Multilayer Perceptron and Radial Basis Function Network); two rule-based systems, C5.0 rules and RIPPER; and an ensemble method, boosted C5.0. Results were obtained with 10-fold cross-validation using default parameters on all algorithms.

At the meta-level we empirically evaluated the k-NN approach to ranking using a leave-one-out method.

5.1 Comparison of Data Characterizations

Figure 1 shows the mean average correlation for increasing number of neighbors obtained by SW ranking method using two different sets of meta-features: the reduced set (Section 2) and an extended set with 25 measures used in previous work [10]. We observe that the results are significantly better with the

⁴ Esprit Long-Term Research Project (#26357) *A Meta-Learning Assistant for Providing User Support in Data Mining and Machine Learning* (www.metal-kdd.org).

⁵ References for these algorithms can be found in [9].

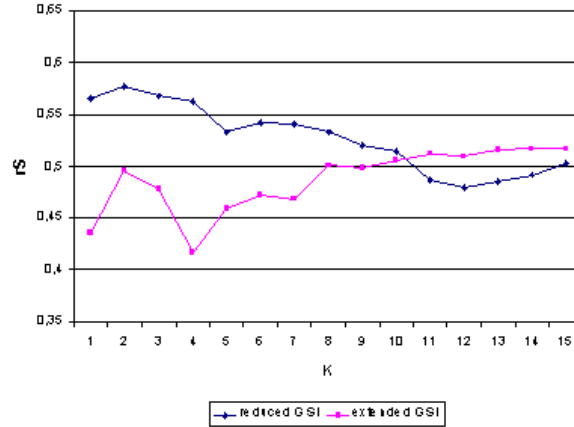


Fig. 1. Mean correlation obtained by SW ranking method for increasing number of neighbors using two sets of GSI data characteristics: reduced and extended.

reduced set than with the extended set. We also observe that the quality of the rankings obtained with the reduced set decreases as the number of neighbors increases. This is not true when the extended set is used. These results indicate that the measures selected do represent properties that affect relative algorithm performance. The shape of the curves also indicates that the extended set probably contains many irrelevant features, which, as is well known, affects the performance of the k-NN algorithm used at the meta-level. Similar results were obtained with the other two ranking methods, AR and SRR.

5.2 Comparison of Ranking Methods

In this section we compare the three ranking methods described earlier for two settings of k-NN on the meta-level, k=1 and 5, using the reduced set of meta-features. The 1-NN is known to perform often well [15]. The 5 neighbors represent approximately 10% of the 45 training data sets, which has lead to good results in a preliminary study [10]. Finally we also evaluated a simple baseline setting consisting of applying the ranking methods to all the training data sets (i.e., 52-NN).

In the next section, we analyse the results of concerning the final goal of providing useful recommendation to the users. But first, we will compare the three ranking methods to each other. We observe in Figure 2 that for k=1, SW obtains the best result⁶. For k=5, AR is the best method and significantly better than the other two, according to Friedman’s test (95% confidence level) and

⁶ As expected, it not significantly different from the other two ranking methods for k=1, because no aggregation is performed with only one data set.

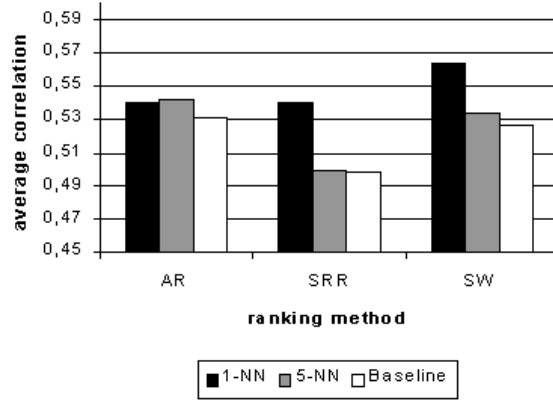


Fig. 2. Comparison of mean average correlation scores (\bar{r}_S) obtained with the 1-NN, 5-NN and the baseline (52-NN) combined with the three ranking methods, AR, SRR and SW.

Dunn’s Multiple Comparisons Procedure (75% confidence level). Comparing the results of the three baselines, we observe that AR is the best at finding a consensus from a set of very diverse rankings. This is consistent with previous results that showed good performance of AR [11]. The results of SRR are somewhat surprising because earlier results in the baseline setting indicated that it was a competitive method [11]. However, the results presented in that paper were based on less meta-data (only 16 data sets).

Comparing these results to the ones presented in the previous section, we observe that the choice of an adequate data characterization yields larger gains in correlation than the choice of ranking method.

5.3 How useful is the recommendation provided?

In this section, we start by comparing the gains obtained with the k-NN approach to ranking when compared to the baseline ranking methods. Next, we take a more user-oriented view of the results, by analysing the trade-off between accuracy and time obtained by the algorithm recommendation method described when compared to cross-validation.

We observe in Figure 2 that meta-learning with k-NN always improves the results of the baseline (52-NN), for all ranking methods. Friedman’s test (95% confidence level) complemented with Dunn’s Multiple Comparison Procedure (75% confidence level) shows that most of the differences are statistically significant. The exceptions are the pairs (1-NN, baseline) and (1-NN, 5-NN) in the AR method and (5-NN, baseline) in the SRR method.

We also observe that there is a clear positive correlation between the recommended rankings generated and the ideal rankings. The critical value for Spearman’s correlation coefficient (one-sided test, 95% confidence level) is 0.5636. Given that we are working with mean correlation values, we can not conclude anything based on this critical value. However, the fact that the values obtained are close to the statistically significant value is a clear indication that the rankings generated are good approximations to the true rankings.

The evaluation performed so far provides information about the ranking as a whole. But it is also important to assess the quality of the recommendation provided by the meta-learning method in terms of accuracy. Since recommendation is provided in the form of a ranking, we don’t know how many algorithms the user will run. We use an evaluation strategy which is common in the field of Information Retrieval, basically consisting in the assumption that the user will run the top N algorithms, for several values of N. This strategy assumes that the user will not skip any intermediate algorithm. This is a reasonable assumption, although, as mentioned earlier, one of the advantages of recommending rankings is that the user may actually skip some suggestions, due to personal preferences or other reasons. In this kind of evaluation, we must take not only accuracy into account but the time required to run the selected algorithm(s). If accuracy is the only criterion that matters, i.e. there are no time constraints, the user should run all algorithms and choose the most accurate.

The cross-validation strategy will be used as a reference to compare our results to. It is the most accurate algorithm selection method (an average of 89.94% in our setting) but it is very time consuming (more than four hours in our setting). As a baseline we will use boosted C5.0, which is the best algorithm on average (87.94%) and also very fast (less than two min.). We also include the Linear Discriminant (LD), which is the fastest algorithm, with an average time of less than five seconds.

The results of the SW method using 1 neighbor and the reduced set of meta-features are presented in Figure 3, assuming the selection of the first 1, 2 or 3 algorithms in the ranking. For each selection strategy (including the baselines), we plot the average loss in accuracy (vertical axis), when compared to CV, against the average execution time (horizontal axis). In the ranking setting, when the algorithm recommended in position N was tied with the one at N+1, we selected, from all the tied algorithms, the ones with the highest average accuracy (in the training data sets) such that exactly N algorithms are executed. The results demonstrate the advantage of using a ranking strategy. Although the Top-1, with an average loss of accuracy of 5.16%, does not seem to be very competitive in terms of accuracy, if the user is willing to wait a bit longer, he/she could use the Top-2 algorithms. The time required is quite good (less than five min., while CV takes more than three hours, on average) and the loss in accuracy is only of 1.23%. Running another algorithm, i.e. running the Top-3 algorithms would provide further improvement in accuracy (1.06% loss) while taking only a little longer.

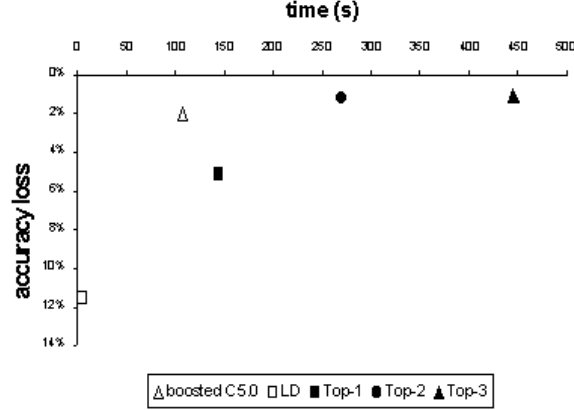


Fig. 3. Evaluation of several algorithm selection strategies (Linear Discriminant, boosted C5.0, Top-1, 2 and 3) according to two criteria (accuracy loss when compared to CV and time). Note that cross-validation takes on average more than three hours.

Comparing to the baselines, we observe that even the Top-1 strategy will be much more accurate than LD but the latter is faster. The comparison of Top-1 with boosted C5.0 is, at first sight, not very favorable: it is both less accurate and slower. However, the Top-2 and Top-3 strategies compete well with boosted C5.0: they are both more accurate but take more time (although, as mentioned above, they still run in acceptable time for many applications).

6 Conclusions

We have investigated different hypotheses concerning the design of a meta-learning method for algorithm recommendation.

First, we compared two sets of data characterization measures. The first is a large set of general, statistical and information-theoretic meta-features, commonly used in meta-learning. The second set was a subset of the first, containing selected measures that represent properties of the data that affect algorithm performance. This selection has significantly improved the results, as would be expected, especially considering that the k-NN algorithm was used at the meta-level. We plan to compare this approach to data characterization with new approaches, like landmarking.

Next, we analyzed a few variants of the recommendation method. We compared two different settings of the k-NN algorithm ($k=1$ and 5) and three different ranking methods to generate a ranking based on information about the performance of the algorithms on the neighbors. We observed that meta-learning is beneficial in general, i.e. results improve by generating a ranking based on the

most similar data sets. The differences in performance between the three ranking methods, although statistically significant in some cases, are not so large as the ones obtained with the selection of meta-features.

Finally, we have compared the results obtained with our ranking approach with the most accurate method for algorithm recommendation, cross-validation (CV) and with boosted C5.0, the best algorithm on average in our set, in terms of accuracy and time. The results obtained show that the strategy of running the Top-2 or 3 algorithms achieves a significant improvement in time when compared to CV (minutes compared to hours) with a small loss in accuracy (approximately 1%). Furthermore, it competes quite well with boosted C5.0, which is faster but less accurate.

References

1. Michie, D., Spiegelhalter, D., Taylor, C.: Machine Learning, Neural and Statistical Classification. Ellis Horwood (1994)
2. Wolpert, D., Macready, W.: No free lunch theorems for search. Technical Report SFI-TR-95-02-010, The Santa Fe Institute (1996)
<http://lucy.ipk.fhg.de:80/~stephan/nfl/nfl.ps>.
3. Brodley, C.: Addressing the selective superiority problem: Automatic Algorithm/Model class selection. In Utgoff, P., ed.: Proceedings of the Tenth International Conference on Machine Learning, Morgan Kaufmann (1993) 17–24
4. Aha, D.: Generalizing from case studies: A case study. In Sleeman, D., Edwards, P., eds.: Proceedings of the Ninth International Workshop on Machine Learning (ML92), Morgan Kaufmann (1992) 1–10
5. Brazdil, P., Gama, J., Henery, B.: Characterizing the applicability of classification algorithms using meta-level learning. In Bergadano, F., de Raedt, L., eds.: Proceedings of the European Conference on Machine Learning (ECML-94), Springer-Verlag (1994) 83–102
6. Atkeson, C.G., Moore, A.W., Schaal, S. In: Locally Weighted Learning. Volume 11. Kluwer (1997) 11–74
7. Lindner, G., Studer, R.: AST: Support for algorithm selection with a CBR approach. In Giraud-Carrier, C., Pfahringer, B., eds.: Recent Advances in Meta-Learning and Future Work, J. Stefan Institute (1999) 38–47
<http://ftp.cs.bris.ac.uk/cgc/ICML99/lindner.ps.Z>.
8. Henery, R.: Methods for comparison. In Michie, D., Spiegelhalter, D., Taylor, C., eds.: Machine Learning, Neural and Statistical Classification. Ellis Horwood (1994) 107–124
9. Pfahringer, B., Bensusan, H., Giraud-Carrier, C.: Tell me who can learn you and i can tell you who you are: Landmarking various learning algorithms. In Langley, P., ed.: Proceedings of the Seventeenth International Conference on Machine Learning (ICML2000), Morgan Kaufmann (2000) 743–750
10. Soares, C., Brazdil, P.: Zoomed ranking: Selection of classification algorithms based on relevant performance information. In Zighed, D., Komorowski, J., Zytkow, J., eds.: Proceedings of the Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD2000), Springer (2000) 126–135
11. Brazdil, P., Soares, C.: A comparison of ranking methods for classification algorithm selection. In de Mántaras, R., Plaza, E., eds.: Machine Learning: Proceedings

- of the 11th European Conference on Machine Learning ECML2000, Springer (2000)
63–74
12. Neave, H., Worthington, P.: *Distribution-Free Tests*. Routledge (1992)
 13. Provost, F., Jensen, D.: Evaluating knowledge discovery and data mining. Tutorial Notes, Fourth International Conference on Knowledge Discovery and Data Mining (1998)
 14. Blake, C., Keogh, E., Merz, C.: *Repository of machine learning databases* (1998)
<http://www.ics.uci.edu/~mlearn/MLRepository.html>.
 15. Ripley, B.: *Pattern Recognition and Neural Networks*. Cambridge (1996)