Visualizing Classifier Performance on Different Domains

Rocio Alaiz-Rodríguez
Dpto. de Ingeniería Eléctrica y de Sistemas
Universidad de León, Spain
rocio.alaiz@unileon.es

Nathalie Japkowicz
School of Information Technology and Engineering
University of Ottawa, Canada
nat@site.uottawa.ca

Peter Tischer
Clayton School of Information Technology
Monash University, Australia
Peter.Tischer@infotech.monash.edu.au

Abstract

Classifier performance evaluation typically gives rise to vast numbers of results that are difficult to interpret. On the one hand, a variety of different performance metrics can be applied; and on the other hand, evaluation must be conducted on multiple domains to get a clear view of the classifier's general behaviour. In this paper, we present a visualization technique that allows a user to study the results from a domain point of view and from a classifier point of view. We argue that classifier evaluation should be done on an exploratory basis. In particular, we suggest that, rather than pre-selecting a few metrics and domains to conduct our evaluation on, we should use as many metrics and domains as possible and mine the results of this study to draw valid and relevant knowledge about the behaviour of our algorithms. The technique presented in this paper will enable such a process.

1 Introduction.

Classifier performance evaluation is a crucial stage in developing machine learning techniques. It typically gives rise to a multitude of results that are difficult to interpret. On the one hand, a variety of different performance metrics can be applied, each adding a little bit more information about the classifiers than the others; and on the other hand, evaluation must be conducted on multiple domains to get a clear view of the classifier's general behaviour.

Caruana et al. [2] studied the issue of selecting appropriate metrics through a visualization method. In their work, the evaluation metrics are classified into three categories and a new metric, SAR, is constructed, that combines the properties found in each of these properties.

In addition to their performance being judged with respect to various metrics, classifiers are often evaluated on several domains that present different characteristics such as dimensionality, types of features, classification difficulty. Japkowicz et al. [4] studied the issue of aggregating the results obtained by different classifiers on several domains. They too use a visualization approach to implement a component-wise aggregation method that allows for a more precise combination of results than the usual averaging or win/loss/tie approaches.

In this research, we take the view that classifier evaluation should be done on an exploratory basis. This can be done manually, though it quickly becomes difficult (if not impossible) to analyze the results manually due to the multitude of metrics and domains considered. Visualization, on the other hand, may be a great aid in this process. Here, we investigate the general issues of visualization and how to adapt existing methods to suit our purpose. This work extends a short paper [1] by illustrating the insights we gain when adopting this approach.

More specifically, we assume that classifier evaluation requires two stages. In the first stage, the researcher computes the results obtained by the various classifiers with respect to several representative metrics on several domains, in order to make the comparison as general as possible. This, of course, will create a considerable amount of data, which will, in turn, need to be analyzed, in a second stage, in order to draw valid and useful conclusions about the algorithms under study. We can say that this second stage is a data mining process in and of itself.

The aim of this paper is to illustrate and motivate the use of graphical methods as a simple alternative approach for the comparison of classifiers. We present a visualization technique (in two dimensions) based on the combination of the above two techniques [2, 4] that allows for analyses with

respect to both a variety of metrics and domains. Note that single scalar metrics can be seen as projections to one dimension. Such projections, however, only allow to show where a classifier stands in relation to *one* other classifier, which usually is the ideal classifier. We will show that our system, based on a projection to two dimensions, allows to study a number of questions that can not be answered with traditional evaluation techniques. Moreover, it provides a way to select candidates for an ensemble of classifiers and enables an analysis from either a classifier point of view or a domain point of view. We illustrate our approach on a study of 15 domains over three representative metrics as per Caruana et al. [2].

The remainder of this article is organized as follows. Sect. 2 discusses the visual approach to analyzing classifier performance, Sect. 3 describes a typical empirical study. Sect. 4 illustrates the benefits of visual data mining in classifier evaluation and Sect. 5 concludes with a summary and suggestions for future work.

2 Visualizing the Classifier Performance

In general terms, classifier performance evaluation involves generating large amounts of performance data and trying to reduce this data to meaningful descriptors of performance. Thus, classifier performance evaluation implies discarding information and data reduction. In this sense, performance evaluation can be approached as a problem of how to project the large amounts of data to a lower dimensional space. Note that in this process, it is desirable to retain as much of the information as possible, discarding only what may be regarded as irrelevant.

In the extreme case, all the performance data gets turned into a single number (projection to one dimension) and the classifiers get compared on the basis of a single quantity, i.e., a scalar metric. However, this involves the maximum amount of information loss and single value indicators of classifier performance are most likely to be unsatisfactory in conveying information about classifier performance. This may be the reason why several single metrics are required to describe different aspects of performance.

In general, the volume of data we need to retain is such that listing numerical values in tables is inadequate and presenting the remaining data in visual form is desirable. Scientific Visualization is a great aid: (a) to carry out data reduction and therefore, communicate what we believe is significant about the performance results and (b) to allow a human observer to easily discover meaningful patterns in the performance results.

In order to compare classifiers on an exploratory basis rather than through standard evaluation, different tools may be useful depending of the amount of data available. They vary from simple approaches to plotting the results in a convenient way (such as histograms, spider graphs, scatter graphs) to dimensionality reduction techniques such as Multidimensional Dimensional Scaling (MDS) [3] or Self Organizing Maps [5]. Although simple graphs are helpful for the analysis, they have limitations as the number of dimensions increases. In this case, a dimensionality reduction technique that preserves the original data structure as much as possible, seems more convenient. Next, we illustrate our approach with a typical empirical study.

3 Typical Experimental Study

Consider a standard empirical study where L classifiers are evaluated on D domains. Consider also that a set of K representative metrics are recorded for each pair of domain-classifier, so that the results can be organized in K tables with elements $m_{ij}^{(k)}$ where k is the metric evaluated, $i = 1, \ldots, L$ and $j = 1, \ldots, D$.

In this section a typical experiment is conducted in order to assess nine classifiers (k-Nearest neighbor with k=1 and k=10 (Ib1, Ib10), Naive Bayes,C 4.5 Decision Tree, Bagged Decision Trees, Boosted Decision Trees, Random Forest, SVM and JRip) on several domains based on three metrics of interest. Evaluation was carried out by 10-fold cross-validation in the WEKA environment [6] with parameters set as default. Fifteen binary classification problems from the UCI repository were assessed in this work (Sonar, Heart-v, Heart-c, Breast-y, Voting, Breast-w, Credits-g, Heart-s, Sick, Hepatitis, Credits-a, Horse-colic, Heart-h, Labor and Krkp). In the following, D1 will refer to Sonar, D2 to Heart-v, and so on.

Different metrics reflect different properties that may be desirable for a classifier. From the three categories established in [2], we chose the most representative ones: RMSE that reflects the classifier's ability to estimate posterior probabilities, AUC with information about its ranking capabilities and the Error Rate metric as a threshold metric. Tables 1, 2 and 3 show the Error rate, RMSE and AUC, respectively for the 15 UCI domains evaluated here.

4 Performance Analysis

Typical questions we would like to answer after the classifier performance analysis is performed are related to similarities/dissimilarities between classifiers: (a) Which classifiers perform similarly so that they can be considered equivalent? (b) Which classifiers could be worth selecting for a classifier ensemble? (c) Does the relative performance of the classifiers change as a function of data dimensionality? (d) Does it change for different domain difficulties?

A first attempt at answering these questions could be to analyze directly the data gathered in Tables 1, 2 and 3. However, such an analysis does not seem straightforward given

Table 1. Error rate for different classifiers on several domains

ERROR RATE															
	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15
Ideal	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ib1	0.1342	0.2957	0.2378	0.2757	0.0986	0.0486	0.2800	0.2481	0.0381	0.1937	0.1884	0.1873	0.2317	0.1733	0.0372
Ib10	0.2402	0.2160	0.1753	0.2699	0.1077	0.0357	0.2600	0.1851	0.0384	0.1737	0.1405	0.1686	0.1660	0.0833	0.0494
NB	0.3211	0.2360	0.1652	0.2830	0.1284	0.0400	0.2460	0.1629	0.0739	0.1554	0.2231	0.2200	0.1629	0.1000	0.1210
C4.5	0.2883	0.2663	0.2248	0.2445	0.0917	0.0544	0.2950	0.2333	0.0119	0.1620	0.1391	0.1470	0.1893	0.2633	0.0056
Bagging	0.2545	0.2513	0.2080	0.2656	0.0895	0.0415	0.2600	0.2000	0.0127	0.1683	0.1463	0.1442	0.2105	0.1533	0.0056
Boosting	0.2219	0.2965	0.1786	0.3035	0.1010	0.0429	0.3040	0.1963	0.0082	0.1420	0.1579	0.1659	0.2142	0.1000	0.0050
RF	0.1926	0.2460	0.1850	0.3144	0.0965	0.0372	0.2730	0.2185	0.0188	0.2008	0.1492	0.1524	0.2177	0.1200	0.0122
SVM	0.2404	0.2463	0.1588	0.3036	0.0827	0.0300	0.2490	0.1592	0.0615	0.1483	0.1507	0.1740	0.1726	0.1033	0.0456
JRip	0.2692	0.2660	0.1854	0.2905	0.0986	0.0457	0.2830	0.2111	0.0177	0.2200	0.1420	0.1306	0.2104	0.2300	0.0081

Table 2. RMSE for different classifiers on several domains

RMSE															
	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15
Ideal	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ib1	0.3512	0.5342	0.3045	0.5042	0.2956	0.1860	0.5278	0.4848	0.1936	0.4252	0.4295	0.4261	0.2950	0.3197	0.1936
Ib10	0.3931	0.4277	0.2179	0.4305	0.2649	0.1519	0.4193	0.3700	0.1699	0.3406	0.3298	0.3587	0.2192	0.3213	0.2458
NB	0.5263	0.4164	0.2256	0.4480	0.3310	0.1945	0.4186	0.3542	0.2285	0.3409	0.4346	0.4179	0.2238	0.1997	0.3018
C4.5	0.5172	0.4531	0.2689	0.4311	0.2760	0.2105	0.4790	0.4526	0.1035	0.3565	0.3290	0.3521	0.2461	0.4209	0.0638
Bagging	0.3926	0.4177	0.2359	0.4335	0.2564	0.1769	0.4201	0.3768	0.0902	0.3388	0.3186	0.3440	0.2290	0.3412	0.0634
Boosting	0.4366	0.4700	0.2497	0.5105	0.2875	0.1864	0.5054	0.4294	0.0757	0.3507	0.3671	0.3690	0.2579	0.2281	0.0603
RF	0.3530	0.4166	0.2295	0.4686	0.2607	0.1615	0.4223	0.3912	0.1156	0.3512	0.3323	0.3376	0.2405	0.2962	0.1116
SVM	0.4837	0.4942	0.2872	0.5470	0.2667	0.1520	0.4979	0.3934	0.2479	0.3606	0.3837	0.4105	0.2885	0.2249	0.2110
JRip	0.4647	0.4360	0.2385	0.4475	0.2828	0.1932	0.44637	0.40846	0.1189	0.4075	0.3419	0.336	0.2574	0.3776	0.0782

Table 3. AUC* (1-AUC) for different classifiers on several domains

AUC* (1-AUC)															
	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15
Ideal	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ib1	0.1361	0.4635	0.2403	0.3687	0.0622	0.0256	0.3400	0.2500	0.1912	0.3362	0.1917	0.2035	0.2512	0.1750	0.0105
Ib10	0.1373	0.4102	0.0920	0.3201	0.0325	0.0759	0.2553	0.1244	0.0672	0.1890	0.0911	0.1366	0.1138	0.0500	0.0094
NB	0.2000	0.2826	0.0955	0.2845	0.0483	0.0120	0.2122	0.0994	0.0747	0.1408	0.1040	0.1501	0.1009	0.0125	0.0479
C4.5	0.2653	0.3983	0.2032	0.3719	0.0629	0.0515	0.3534	0.2450	0.0505	0.3034	0.1064	0.1507	0.2341	0.2666	0.0012
Bagging	0.1478	0.2869	0.1296	0.3518	0.0362	0.0105	0.2469	0.1291	0.0050	0.1769	0.0771	0.1237	0.1178	0.1583	0.0007
Boosting	0.0938	0.3055	0.1187	0.3569	0.0370	0.0176	0.2770	0.1166	0.0123	0.2003	0.0945	0.1118	0.1389	0.0625	0.0007
RF	0.0889	0.2914	0.1215	0.3537	0.0376	0.0137	0.2499	0.1386	0.0072	0.1599	0.0886	0.1023	0.1444	0.0916	0.0012
SVM	0.2418	0.4335	0.1639	0.4072	0.0869	0.0316	0.3292	0.1633	0.5001	0.2487	0.1434	0.1912	0.2033	0.1250	0.0457
JRip	0.2631	0.4366	0.1591	0.3877	0.0839	0.0368	0.3871	0.2041	0.0579	0.3960	0.1285	0.1562	0.2427	0.2416	0.0055

the quantity of results recorded (and there could be worse instances of this).

As an alternative, metrics like SAR try to summarize all the gathered information with a point estimation. Thus, SAR carries out the projection

$$\mathrm{SAR}^* = 1 - \mathrm{SAR} = \mathrm{RMSE} + \mathrm{Error} + \mathrm{AUC}^* \qquad (1)$$

where $\mathrm{AUC}^* = 1 - \mathrm{AUC}$. The closer to zero the SAR[1] values (and all its components) are, the better the classifier performs. Tables 4 shows the classifiers' performance values according to the SAR metric and the individual metrics involved in its definition computed as the average over all the domains. Based on these metrics, a ranking can be established among classifiers (see Table 5). This ranking suggests, for instance, that two classifiers that are close enough to each other may be considered equivalent or related by similar performance properties. However, such conclusions are sometimes misleading and it is more beneficial to extract more information by exploring the results visually.

[1]Note that this metric lies in the interval [0,1].

4.1 Visualization and exploration based on a MDS projection

In this section, we demonstrate the use of MDS (Multi-Dimensional Scaling) [3], to visualize either the classifiers or the domains in a graph, so that interpoint distances in the high dimensional space are preserved as much as possible in the 2D space.

Let us now study what information may be extracted from a graph in which the information provided in Tables 1, 2 and 3 is not simply averaged (over domains and over different metrics) but is projected using MDS. The distance between two points is calculated as the Euclidean distance and the stress criterion is normalized by the sum of squares

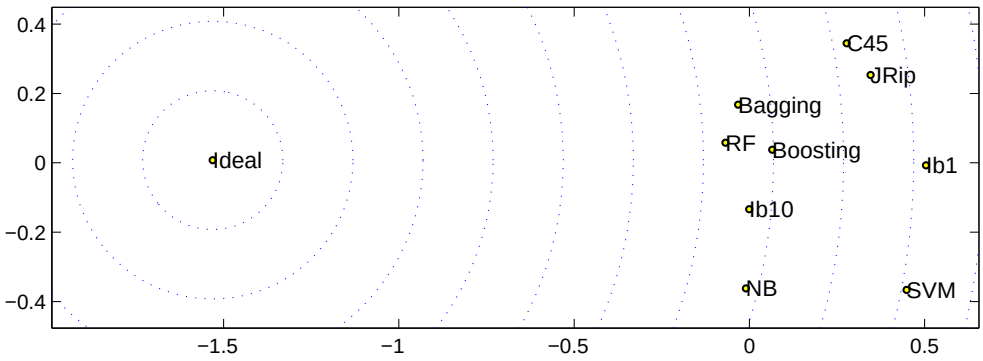


Figure 1. metric MDS projection from 45 dimensions to 2 dimensions based on the RMSE, AUC* and Error rate gathered over 15 domains.

Table 4. Classifier evaluation according to different performance metrics according to the averages over all the domains.

	CLASSIFIER EVALUATION METRICS			
	AUC*	RMSE	Error Rate	SAR* (1-SAR)
Ideal	0.0000	0.0000	0.0000	0.0000
Ib1	0.2164	0.3647	0.1779	0.2530
Ib10	0.1358	0.3107	0.1539	0.2001
NB	0.1244	0.3375	0.1759	0.2126
C4.5	0.2043	0.3307	0.1744	0.2365
Bagging	0.1332	0.2957	0.1607	0.1965
Boosting	0.1296	0.3190	0.1625	0.2037
RF	0.1260	0.2992	0.1623	0.1958
SVM	0.2210	0.3499	0.1551	0.2420
JRip	0.2125	0.3224	0.1739	0.2362

Table 5. Classifier ranking according to different performance metrics.

	CLASSIFIER RANKING			
	AUC*	RMSE	Error Rate	SAR* (1-SAR)
1	Ideal	Ideal	Ideal	Ideal
2	NB	Bagging	Ib10	RF
3	RF	RF	SVM	Bagging
4	Boosting	Ib10	Bagging	Ib10
5	Bagging	Boosting	RF	Boosting
6	Ib10	JRip	Boosting	NB
7	C45	C4.5	JRip	JRip
8	JRip	NB	C4.5	C4.5
9	Ib1	SVM	NB	SVM
10	SVM	Ib1	Ib1	Ib1

of the interpoint distances.

Before starting to explore the graphical representation, it is interesting to assess the stress criterion. In other words, it is important to know how much of the original data structure is preserved after projecting the data to two dimensions. We can then have an idea of the information gained when moving from one dimensional representation to two dimensions. In our example the stress becomes 0.08 for two dimensions (not much loss of information), but it increases to 0.22 when considering only one dimension.

Each classifier-i is originally described by a vector with components $m_{ij}^{(k)}$, where $k = 1, \ldots, K$ and $j = 1, \ldots, D$. ($D = 15$ and $K = 3$, which leads to a 45 dimensional space). After projecting to 2 dimensions with MDS, each classifier-i is described by a point (x_i, y_i) and the similarities/dissimilarities among classifiers can be analyzed in Fig.1, which gives a clearer view of the relative performance between classifiers. The ideal classifier is also included, which allows us to compare classifiers by their projected distance to the ideal classifier as well as their relative position to other classifiers.

Next, we will show that our system allows us to study a number of other questions that cannot normally be answered with traditional evaluation tools. These may refer to an analysis from a classifier point of view or from a domain point of view. In the former, the objects are the classifiers (domains are reduced when projecting) and in the latter the objects are the domains (classifiers are are the attributes projected).

- **Classifier-Centric Questions:**
  - Can the classifiers be organized into equivalence classes that perform similarly on a variety of domains?
  - In what way are the classifiers similar or different from one another?
  - Which classifiers would it be beneficial to combine in an ensemble of classifiers? Which combinations would not improve the results?
- **Domain-Centric Questions:**

- Can domains be organized into equivalence classes within which various classes of classifiers behave predictably?
- What domain characteristics influence the behaviour of different domains (e.g., domain difficulty, dimensionality, etc.)?

4.1.1 Classifier similarity/dissimilarity

In this section we analyze the data performance from a classifier point of view.

Classifier clustering

From the point metric SAR (see Table 4), we can easily draw the conclusion that C4.5 and SVM performance are very similar. The same applies to C4.5 and JRip and also to Boosting and NB.

However, in the projection that SAR represents (from 45 dimensions to 1), we lose a lot of information about the similarities between classifiers. Keeping more information (projecting to 2 dimensions) allow us to identify several clusters of classifiers whose performance are very close or equivalent across the fifteen domains in terms of the three metrics considered (RMSE, AUC and Error rate). With the aid of Fig. 1 the two following classifier clusters with equivalent performance can be identified: {C4.5, JRip} and {Bagging, RF, Boosting}.

If we now go back to Tables 1, 2 and 3, we would be able to confirm the similarities among the classifiers within the cluster. Nonetheless, finding the similarities directly from the information gathered in these tables does not seem straightforward.

Recalling the similarities found by analyzing the SAR metric, we are able to conclude that: (i) C4.5 and JRip's performance are very close (this corroborates SAR-based analysis) , (ii) C4.5 and SVM's performance are divergent (although their difference to the ideal classifier seems to be approximately equal) and (iii) Boosting and NB's behaviours are not as close as the information in Table 4 suggests. While this clarifies the results, it also suggests a whole series of new questions: In which way are these classifiers different? Where do these differences among classifiers come from? Do they arise, for example, because of different capabilities to estimate posterior probabilities? Can we impute them to the domain characteristics? In that case, which characteristics of the domain make these differences appear?. Next, we analyze these in more detail.

Classifier dissimilarities

We will now explore whether the dissimilarities between various classifiers are due to differences in certain aspects of performance (ranking properties, posterior probability estimation, accuracy). Let us focus, again, on the two pairs of classifiers which the SAR-based analysis found similar but which the general MDS projection showed to be quite different: Boosting/NB, C4.5/SVM.

We consider a specific metric $k = k_l$ and a pair of classifiers. Each classifier-i is described by a vector with components $m_{ij}^{(k_l)}$, where $j = 1, \ldots, D$. (space with 15 dimensions since $D = 15$). Fig.2 shows the similarities/dissimilarities among the classifiers in terms of AUC, RMSE and Error. We can analyze whether or not they are similar in any of the properties that these metrics reflect.

- *C4.5 vs. SVM*.
  Fig.2(a) shows that the absolute difference in terms of AUC with respect to the ideal classifier is approximately equal for both classifiers. However, C4.5's performance is quite far from that of SVM. The same applies to the analysis in terms of RMSE (Fig.2(b)) and accuracy (Fig.2(c)). Therefore, both classifiers are very different in all the aspects of performance analyzed here. This leads us to conclude that the domain characteristics play an important role. There may be certain domains for which C4.5 clearly outperforms SVM and viceversa.

- *Boosting vs. NB*. The plots as a function of the individual metrics do not show closeness between classifiers for any of them.

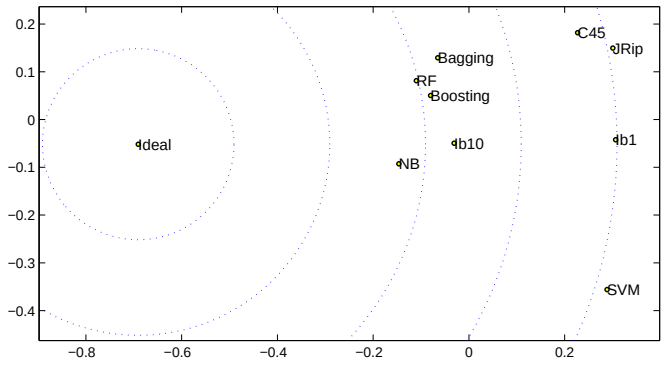
From this analysis we can conclude that these two pairs of classifiers are not similar in any of the relevant performance characteristics we are evaluating. They show very different behaviour across domains, although in absolute terms (with respect to the ideal classifier) they give the same overall performance. An analysis from the domain point of view will probably provide information about the origin of the differences (see Section 4.1.2).

Ensembles of classifiers

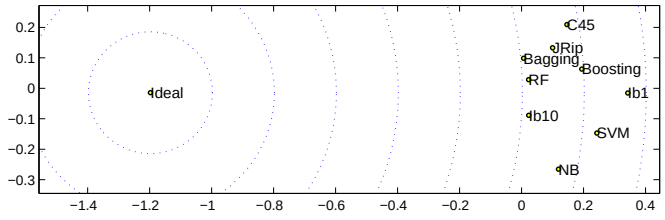
The aim of the classifier ensembles is to take advantage of the individual classifiers' capabilities by selecting or weighting their individual decisions. Here, we analyze performance across domains and therefore, the discussion about possible combinations is taken with regard to the domain the ensemble has to classify instances from. Nonetheless, this analysis can be carried out at the instance level as well and be used to evaluate possible combinations of classifiers to classify test examples of a given domain.

If, given a domain, we could predict which classifier from a set of decision machines gives the best performance, or how to combine them in order to exploit the individual decisions, we would be able to develop a general classifier with better general performance.

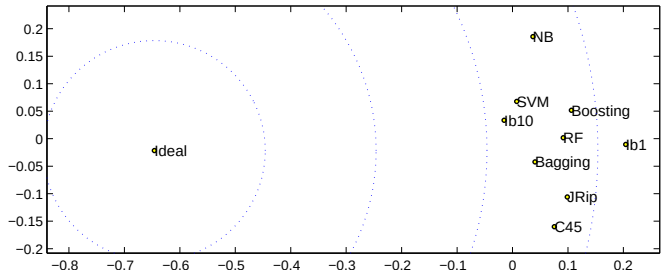
Our tool helps us see what possible combinations we may explore. The dotted lines in Fig.1 can be viewed



(a)



(b)



(c)

Figure 2. metric MDS projection from 15 dimensions to 2 dimensions: (a) based on AUC. (b) based on RMSE. (c) based on Error rate.

as iso-absolute-performance curves. Their points, though, have very different relative performance. We consider as possibly good combinations those classifiers that are in the same iso-absolute-performance curve and are far from each other. Note that this tool allows to identify combinations like {Boosting, NB, Ib10} or {C4.5, Ib1, SVM} that could be worth exploring. Developing combination schemes, though, is beyond the scope of this work and will be considered in future research.

4.1.2 Classifier relative performance across domains

When using performance metrics to look at how classifiers A and B perform on different domains, we can treat each domain as a point in a higher dimensional space. For each performance metric, we evaluate the metric with respect to classifier A and with respect to classifier B and then, take the difference. Thus, each domain-j is represented by a vector with components $(m_{Aj}^{(k)} - m_{Bj}^{(k)})$ where $k = 1, \ldots, K$. For instance, if we are using 3 performance metrics, AUC, RMSE and Error, we get 3 attribute values for each domain. We can then use MDS to project this to 2 dimensions. This way, we will be able to observe the domains distributed according to how close/different they are with regard to the relative performance between the classifiers. Two domains will be plotted close to one another, if the relative performance between the classifiers is close in these domains.

We illustrate this study in Fig.3 with the classifiers C4.5/SVM that were previously found to have similar absolute performance but turned out to be quite different from one another. For example, we can observe that in Domains D9 and D14 the relative performance of these two classifiers is very different (in D9, C4.5 clearly outperforms SVM whereas SVM outperforms C4.5 in D14). In the group {D4, D15}, C4.5 is also preferred to SVM, but the relative difference changes in domain D9. In other groups, however, the differences may be blurred, but there are still domains where the classifiers have the same relative performance. Thus, in {D3, D10, D7, D13}, SVM is better in terms of Error and AUC but it is worse when assessed for RMSE.

Identifying to which group a domain belongs to, could allows us to predict which classifier of the two would be preferred for that problem.

4.1.3 Domain similarity/dissimilarity

Apart from a general study about classifiers, our tool provides a simple way to analyze classifier performance ac-

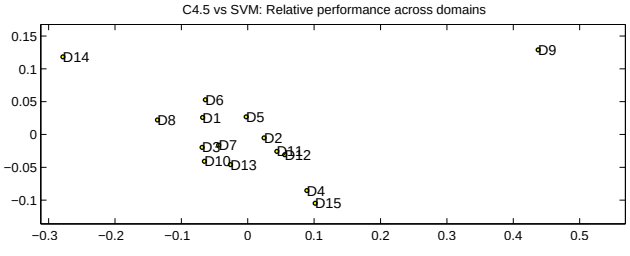


Figure 3. MDS projection with relative performance information (AUC, RMSE, Error differences) between SVM and C4.5 across domains (projection from 3 to 2 dimensions).

cording to the domain characteristics.

As previously mentioned, we can regard each domain as an object with attributes representing a measure of how several classifiers have performed on that domain. Note that the attributes are classifier performance measures and the classifier dimensions are the ones reduced.

Domain clustering based on task difficulty

It is possible to identify groups of domains according to how the classifiers behave on these. In this analysis, each object-j is a domain defined by a vector with components $m_{ij}^{(k_l)}$ with metric $k = k_l$ and classifiers $i = 1, \ldots, L$.

For instance, let us focus on the classifier ranking properties (measured by the AUC metric). Our original space has 16 objects (15 domains plus an ideal one, D0, for which all classifiers get the maximum AUC) and has 9 dimensions (as many as classifiers; L=9). There are domains for which ranking becomes easier and others for which it is a more difficult task. We argue that the classifier performance may differ according to the domain difficulty.

In Fig.4(a) where the domains are represented, we can see that for domains such as D6, D5, D15 the ranking is easier than in others like D7, D4 and D2.

Fig.4(b), Fig.4(c) and Fig.4(d) show the classifier relation based on AUC for simple, medium and high difficulty domains, respectively. These graphs may allow us to extract information with regard to domain difficulty and draw conclusions such as the following: "As domain difficulty increases, classifier A becomes less competitive than classifier B..."

For example, from Fig.4 we can reach the following conclusions in terms of classifier ranking capabilities:

1. NB tends to improve as difficulty increases. In fact, it appears as the best alternative for scenarios where ranking is difficult.
2. Differences between Bagging and Boosting seem to widen as difficulty increases. As a result, in difficult domains Boosting performance is much worse than Bagging.
3. The classifier with best general performance across difficulty is RF.

While the ranking established in Table 5 identifies NB as the the best paradigm for ranking purposes, this tool allows us to see that great improvements can be achieved choosing the decision system depending on the domain difficulty.

Domain clustering based on the Input Space Dimensionality

Another important aspect in classification is the dimensionality of the input space. Next, we explore the classifier ranking properties according to the dimensionality of the input space. We focus on a set of domains that fulfill certain restrictions and describe them by the components $m_{ij}^{(k_l)}$ with metric $k = k_l$ and classifiers $i = 1, \ldots, L$ (each object is described in a $L = 9$ dimensional space).

Fig.5 shows the classifier proximity in terms of AUC for domains with low dimension (lower than 20) and higher dimensionality. It is straightforward to see that the relative distances between the classifiers change as well as their distance to the ideal solution.

We can see that for low dimension domains, RF, Bagging and Boosting are relatively close to one another, but that they are outperformed by the NB classifier. However, as dimensionality increases, RF outperforms the remaining classifiers. In this case, the analysis reveals the optimality of the classifier depending on the size of the input feature space.

5 Conclusions and further research

In this work, we take the view that classifier comparison should be done on an exploratory basis rather than through standard evaluation. This means that as long as the performance analysis progresses, we will discover tendencies, similarities, dissimilarities or outliers. There is no need to know what we want to find in advance.

We provide a technique based on visualization that takes this very general view and transforms it into a practical endeavour. We show that visual data mining is a powerful tool for discovering data patterns, a task that is quite difficult when simply looking at the results organized in tables, and inaccurate when summarized by a SAR-like measure. Moreover, it allows us to analyze the performance data not only from a classifier point of view, but also from a domain point of view.

In this work we began by illustrating how to combine several metrics that are recorded for several scenarios. We found out that the conclusions drawn based on point metrics are sometimes misleading and too simplistic. A deeper analysis allowed us to understand the underlying relation between classifiers and domains.

There are many more avenues to explore in model selection and combination tasks. Assessing the benefits of this tool to select classifiers in an ensemble is our immediate future research.

References

[1] R. Alaiz-Rodriguez, N. Japkowicz, and P. Tischer. A visualization-based exploratory technique for classifier comparison with respect to multiple metrics and multiple domains. In *Proceedings of the 2008 European*

Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, 2008.

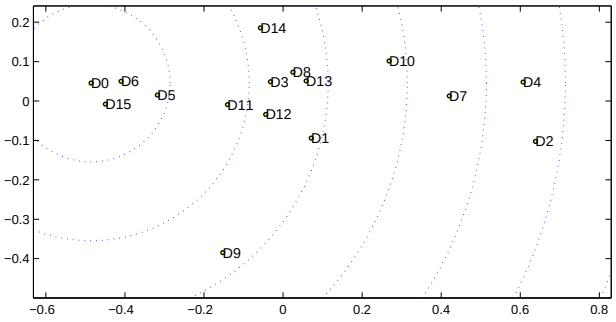
[2] R. Caruana and A. Niculescu-Mizil. Data mining in metric space: An empirical analysis of suppervised learning performance criteria. In *Proceedings of the Tenth International Conference on Knowledge Discovery and Data Mining (KDD'04)*, 2004.

[3] T. Cox and M. Cox. *Multidimensional Scaling*. Chapman and Hall, October 1994.

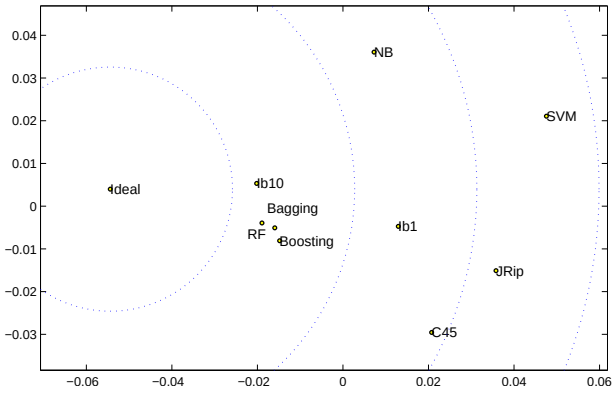
[4] N. Japkowicz, Sanghi, and P. Tischer. A projection-based framework for classifier performance evaluation. In *Proceedings of the 2008 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2008.

[5] T. Soukup and I. Davidson. *Visual Data Mining: Techniques and Tools for Data Visualization and Mining*. Wiley, October 2002.

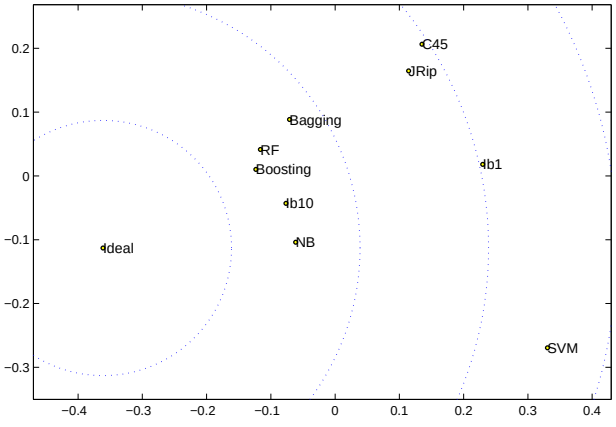
[6] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, October 1999.



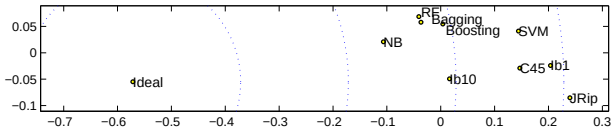
(a)



(b)

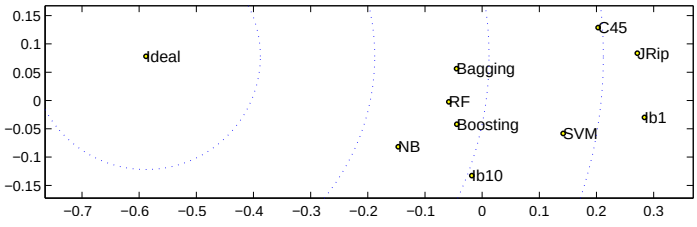


(c)

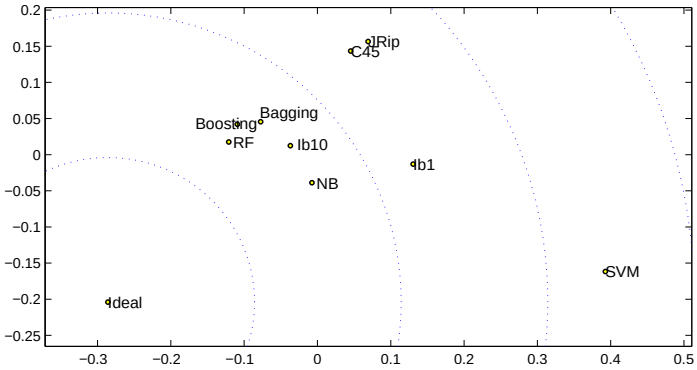


(d)

Figure 4. Based on AUC.(a) Domain Complexity. (b) Simple domains (5, 6 15). (c) Domains (14 11 12 3 8 13 1 9).(d). Complex domains (10 7 4 2).



(a)



(b)

Figure 5. MDS based on AUC (from 15 to 2 dimensions).(a) Low dimensionality of the input space (lower than 20) (b) Higher dimensionality of the input space