Classifier Utility Visualization by Distance-Preserving Projection of High Dimensional Performance Data

Nathalie Japkowicz*
School of Information Tecnology and Engineering
University of Ottawa
Ottawa, Canada

Pritika Sanghi and **Peter Tischer**
Clayton School of Information Technology
Monash University
Melbourne, Australia

Abstract

In this paper, we propose approaching the problem of classifier evaluation in terms of a projection from a high-dimensional space to a visualizable two-dimensional one. Rather than collapsing confusion matrices into a single measure the way traditional evaluation methods do, we consider the vector composed of the entries of the confusion matrix (or the confusion matrices in case several domains are considered simultaneously) as the performance evaluation vector, and project it into a two dimensional space using a recently proposed distance-preserving projection method. This approach is shown to be particularly useful in the case of comparison of several classifiers on many domains as well as in the case of multiclass classification. Furthermore, by providing simultaneous multiple views of the same evaluation data, it allows for a quick and accurate assessment of classifier performance.

1 Introduction

Performance evaluation in supervised classification is traditionally performed by considering the confusion matrices obtained from test runs of several classifiers on various domains, collapsing each matrix into a value (e.g., accuracy, F-measure), and comparing these values to each other. One issue with this approach is that, by the time the classifiers' performances get compared to one another on a given domain, the details of the confusion matrices have been lost. The comparison only involves a single number, be it the accuracy or F-measure of the classifiers. The problem is compounded if the comparison involves several domains, and, when dealing with multi-class rather than binary domains.

In order to defray this problem, people sometimes use pairs of values on which to base their comparisons. Precision/Recall and Sensitivity/Specificity are two commonly used pairs. While this alleviates the problem, somewhat, it makes the comparison of classifiers more complex since it creates cases where one classifier obtains good results on one component and bad ones on the other, while the second classifier obtains opposite results. Furthermore, such pairs of values do not apply to multi-class domains, and the problem of how to aggregate the results obtained on various domains remains as well.

The purpose of this paper is to propose a different way to view the performance evaluation problem with the hope of addressing these issues while offering a more generalized vision of the evaluation problem. In particular, we can view classifier evaluation as a problem of analyzing high-dimensional data, recognizing that the performance measures currently used by the data mining community are but one class of projections that could be applied to these data. If we think of our current measures as specialized projection methods, we can then generalize the procedure by considering the fact that any projection method (standard or not) could be applied to our highly dimensional performance data, along with any distance measure (once again, standard or not). Such an approach could open up the field of classifier evaluation by allowing us to both organize and classify the existing measures within this new framework and, more importantly, to experiment with a variety of new approaches in a more systematic way. A particular benefit of this framework is the fact that projection approaches are typically intended for visualization, which is useful in that it permits both a quick assessment of the results by a human-being and the compounding of more information into the representation than in the case where a single or a pair of values are issued. This, by the way, is in line with more recent evaluation methods such as ROC Analysis (Fawcett 2003) and cost-curves (Drummond & Holte 2006) which also suggest a move towards visual approaches.

The research presented in this paper demonstrates the kind of classifier performance evaluation strategies that can be derived from the consideration of this generalized framework. This paper focuses on three particular advantages brought on by our new vision: its solution to the aggregation of results on different domains; its approach to dealing with multiclass domains; and the fact that it permits the quick and easily interpretable generation of multiple views of classifier performance. Please note that this paper restricts itself to a small number of options with respect to the projection approaches, distance functions and result data representation that could be used, with the understanding that future work will explore these possibilities further. It is also important to note that although we focus on the evaluation of supervised classification algorithms, here, our approach is universal and could be applied to any performance evaluation problem domain.

*This work was done at Monash University while on sabbatical.

The remainder of the paper is organized as follows: Section 2 details our framework and its particular implementation we adopted in this paper. In particular, we discuss the kind of performance data we use as a starting point, the distance measures considered, as well as the various projection methods we evaluated. The purpose of Sections 3 and 4 is to demonstrate the aggregation properties of our framework. In particular, Section 3 illustrates our approach in the case where a number of classifiers are compared on several domains simultaneously while Section 4 considers the case where the same classifiers are compared on a single multi-class problem. In both sections, we highlight the particular advantages of our technique. Section 5 discusses how our method can be used as a multi-facetted approach to classifier performance evaluation. Section 6 concludes the paper, and discusses potential extensions for future work.

2 The Framework and its Implementation

As discussed in the introduction, current evaluation methods can be viewed as specialized projections from a high-dimensional to a 1-dimensional space, in the case of Accuracy, F-Measure and AUC, and to a two-dimensional space, in the case of Precision/Recall and Sensitivity/Specificity. In this work we generalize this idea by suggesting that the techniques proposed in the field of visualization can be put to the service of classifier evaluation as well. In particular, we propose to use the projection techniques and distance measures in use in that field for our purpose. We begin by discussing the general methodology we adopted, and then move on to addressing the issue of choosing an appropriate projection method.

2.1 General Methodology

The visualization approach we propose works according to the following steps:

1. All the classifiers involved in the study are run on all the domains considered, and the corresponding performance matrices (be they confusion matrices, performance vectors of the outcome on each testing point, etc...) are saved.
2. The performance matrices associated with one classifier on each domain are organized into a single vector. The process is repeated for each classifier such that there is a pairwise correspondence of each vector component from one classifier to the next one.
3. A distance measure is chosen to represent the distance between two vectors in high-dimensional space.
4. A projection method is chosen to project the vectors into a two-dimensional space.[1]
5. The distance measure and projection method are used on the vectors generated in Step 2.

The traditional approach to classifier performance evaluation is compared to our new approach in Figure 1. As shown in that figure, in the traditional approach, the performance value of a classifier is calculated on each domain, be

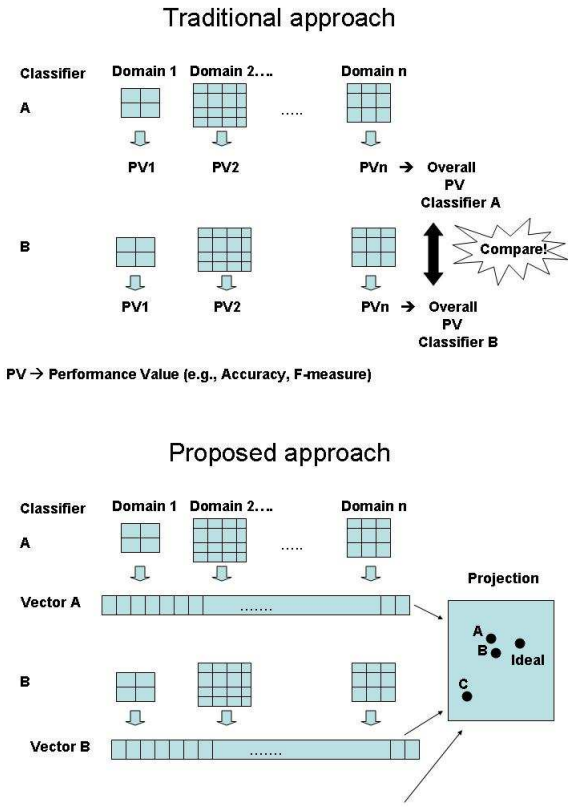

Figure 1: The Traditional and Proposed Approaches to Classifier Performance Evaluation

it binary or multiclass. These values are then aggregated together into an overall performance value, that gets compared from classifier to classifier. In the new approach, the data pertaining to a classifier is preserved into its original form and simply concatenated into a vector. The transformation is delayed until the projection is applied. This means that in our approach, information is lost in a single spot: the projection phase. In the traditional approach information is lost whenever any kind of aggregation occurs.[2]

If we consider the performance of several classifiers on a single binary domain, there are two advantages provided by our new framework. First, it decomposes the problem in a principled manner, separating the issue of projection from that of choosing an appropriate distance measure along which to compare the data. Secondly, by going from a projection to a one-dimensional space to a projection to a two-dimensional one[3], it allows for two rather than one relation-

[1] Three or four dimensions could also be used, if that could be helpful.

[2] Note, however, that since, in both the traditional approaches and our approach, as considered in this paper, we take as a starting point the confusion matrix—an aggregated form of result—, information has been lost even before either performance evaluation approach is used.

[3] Even though the Precision/Recall and Specificity/Sensitivity approaches allow for a two-dimensional projection, the projected

ships to be established. In the traditional approach which, typically, projects the performance data into a single dimension, the classifiers can only get ranked according to their similarity to the ideal classifier. In our evaluation framework, the addition of a dimension allows the classifiers not only to be ranked according to the ideal classifier, but also, to be compared to one another.

A third key advantage over the traditional approach concerns the aggregation of classifier results over different domains. It is common for researchers to simply average the results obtained by a classifier over different domains. This is a mistake when dealing with multi-class classification problems since the same value has different meanings depending on the number of classes. Recognizing this problem, researchers sometimes use a win/tie/loss approach, counting the number of times each classifier won over all the others, tied with the best or lost against one or more. This approach, however, loses any kind of information pertaining to how close classifiers were to winning or tieing. Our approach does not suffer from either of these problems since the entries of each performance vector are compared, in a pairwise fashion, from classifier to classifier.

Please, note that if an unweighted distance measure is used in the projection method, then each matrix entry is given the same importance, but this can be changed by weighting the measure appropriately.

2.2 Implementation Details

Several points considering the implementation of our approach need to be clarified. First, it is important to note that the vectors representing each classifier can take different formats. They can, simply, be 4-dimensional vectors containing all the entries of the confusion matrix on a single binary domain, 9 dimensional vectors containing all the entries of the confusion matrix on a single 3-class domain, and so on. As well, they can be formed by the confusion matrices obtained by a single classifier on several domains, be they multi-class or binary domains. It is also possible, rather than representing the confusion matrices, to represent the classifiers' outcome on each point of the testing set. The graph of Figure 2 is an example where such a representation was used. It plots the combined outcome of eight classifiers on three UCI domains: Breast Cancer, Labour and Liver. In particular, it takes into account the classification of all the data in the three training sets since we ran 10-fold crossvalidation so that each point appears in one of the testing folds). Since Breast-Cancer contains 286 instances, Labour, 57, and Liver 345, each vector in the original data set from which the projection is plotted has dimensionality 688. Alternatively, in the graph of Figure 3 of Section 3, since only the confusion matrices of the three binary domains are considered, the original data set from which the projection is plotted has dimensionality 12.

Second, we must specify what distance measure and projection approach we selected for implementing the method. The distance measures can take several forms, each with different properties. The Euclidean distance (L2 norm), for example, considers all the performance data equally, though it penalizes more for the presence of a few extreme differences than for the presence of several small differences. The Manhattan distance (L1 norm) attaches less importance to large differences. Other distance measures can weigh different components differently. For example, true positives can be given more importance than true negatives (similarly to Precision). In a multi-class domain, a distance measure can focus on the performance of one class, grouping all the other classes, and so on. In fact all the biases provided by the traditional measures (accuracy, precision, recall, F-measure and so on) can be reproduced in our framework. In our particular study, the main distance function we will consider is the Euclidean distance. However, the Manhattan Distance, as well as a measure that provides an emphasis on one class versus all the others, will be discussed briefly in Section 5.

Third we must discuss our choice of a projection approach. In this work we considered two methods: Principal Component Analysis (PCA)(Jain, Duin, & Mao 2000)[4], a linear projection, and a non-linear distance-preserving projection approach, recently proposed by (Lee, Slagle, & Blum 1977; Yang 2004). The second approach, in addition to being non-linear, was considered because it has the advantage of guaranteeing that the distance from one point to at least one of its nearest neighbours is preserved. Our decision as to which projection to use for this paper was based on whether or not there was a need to use a non-linear method. In particular, we plotted a number of graphs using the two projections and compared their results. In many cases, the linear and non-linear projections yielded similar information, but there were a few cases were the outcome of the non-linear approach was more reasonable (when compared to the results obtained with traditional evaluation measures). For example, the plot of Figure 2 presents the PCA projection of the outcome of the classification by eight different classifiers (please, see below) on all the data contained in three UCI (C.Blake & Merz 1998) domains: `Breast Cancer`, `Labour` and `Liver`. The information provided in this plot is obviously misleading since classifier *IBk*'s closeness to the ideal classifier (both are located on the x-axis) is not warranted. This can be seen in Table 5, later on in the paper, where it is clear that *Ibk* does not display any behaviour distinguishing it particularly favourably from the other classifiers.[5]

In view of these results we decided to adopt the non-linear distance-preserving projection for all our remaining experiments, in order to improve our chances of projecting more accurate information in all cases. The detailed description of this projection method follows in the next subsection. PCA and MDS are not described since they were not adopted and since they are well-known projection approaches.

data is typically treated as two 1-dimensional projections rather than one 2-dimensional projection.

[4] PCA is equivalent to Multi-Dimensional Scaling (MDS)(Jain, Duin, & Mao 2000) in our setting since the use of the Euclidean distance makes the results of the two approaches indistinguishable.

[5] In the corresponding non-linear plot, each classifier is placed at roughly the same distance to Ideal. This is more reasonable than the information suggested by the PCA/MDS graph. The non-linear plot is not shown here because of space concerns.

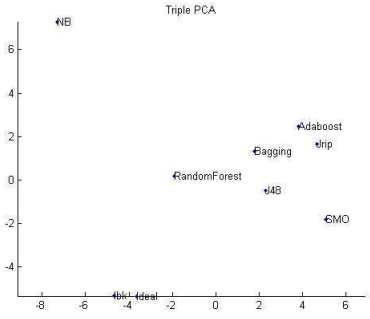

Figure 2: The PCA/MDS projection of three binary domains represented by the outcome of the classifiers on each data point

2.3 A Distance-Preserving Projection Approach

Our approach is a slight variation on an approach by (Lee, Slagle, & Blum 1977; Yang 2004). It is described as follows:

Let $d(x, y)$ represent the distance between x and y in the original higher dimensional space; let $P(x)$ and $P(y)$ be the projections of x and y onto the two-dimensional space; and let $d_2(P(x), P(y))$ represent the distance between the projected points in a two-dimensional space. In this case, we are projecting the performance of the classifiers, p_i where $i = 1, 2, \ldots n$. We introduce the ideal classifier as p_0. p_0 is mapped to the origin.

Find the classifier which is closest to ideal, p_1, and put this on the y-axis at $(0, d(p_0, p_1))$.

For the remaining classifiers, at each stage we find the classifier, p_i, which is nearest to the classifier which has just been plotted, p_{i-1}. When plotting p_i we want to preserve two constraints:

$$d_2(P(p_i), P(p_{i-1})) = d(p_i, p_{i-1}) \quad (1)$$

i.e. we want the projections of p_i and p_{i-1} to be the same distance apart as p_i and p_{i-1}.

We also want to satisfy the second constraint:

$$d_2(P(p_i), P(p_0)) = d(p_i, p_0) \quad (2)$$

i.e. we want the projection of the ideal classifier and the projection of p_i to be the same distance apart as the classifiers are. This means that in the projected space the distance from the origin is a measure of how close the classifier is to ideal. The better the classifier, the closer its projection will be to the origin.

Most times there will be two possible positions for $P(p_i)$ which satisfy both constraints. When there is a choice of solutions, the solution is chosen to satisfy a third constraint as closely as possible:

$$d_2(P(p_i), P(p_{i-2})) = d(p_i, p_{i-2}) \quad (3)$$

Whereas we choose p_i to be the point which has not yet been projected which is closest to the most recently projected point, the original algorithm by (Lee, Slagle, & Blum 1977; Yang 2004) chooses p_i to be the point which has not yet been projected and which is closest to *any* of the points which have already been projected. The original approach projects the points in the same order as Prim's algorithm would add the points to a Minimal Cost Spanning Tree. Both approaches were tried, but we preferred the results produced by the modified approach because it seemed to separate clusters more.

Please, note that in our graphs we have found it useful to draw lines between pairs of projected points to show that the distance between the projected points is equal to the distance between the points in the original, higher dimensional space. Dotted lines connect projected points to the original and indicate the exact distance in the higher dimensional space from the classifier to the ideal classifier. Unbroken lines connect a point to the point that was projected immediately before it in the projection order. The distance between these projected points is also identical to the distance between the points in the original space.

When looking at the projected points, it is useful to remember that the triangle formed by $P(p_0)$, $P(p_{i-1})$ and $P(p_i)$ is congruent to the one formed by p_0, p_{i-1}, and p_i.

3 Experiments on Multiple Binary Domains

In this part of the paper, we experiment with the use of our approach on multiple domains. The three domains considered are all from the UCI Repository for Machine Learning and are: `Breast Cancer`, `Labour` and `Liver`. This means that we are projecting vectors of size 12 (3 confusion matrices of 4 entries each) into a two dimensional domain. Eight different classifiers were compared in this study: *Naive Bayes (NB), C4.5 (J48), Nearest Neighbour (Ibk), Ripper (JRip), Support Vector Machines (SMO), Bagging (Bagging), Adaboost (Adaboost)* and *Random Forests (RandFor)*. All our experiments were conducted using Weka (Witten & Frank) and these particular algorithms were chosen because they each represent simple and well-used prototypes of their particular categories. The results we report were obtained using 10-fold stratified cross-validation. It is worth noting that since the purpose of all our experiments was to interpret the results produced by our evaluation method and not to optimize performance, default settings of Weka were used throughout the paper. The significance of this work, thus, does not lie in the results we obtain, which should only be seen as illustrative of the evaluation framework we propose, but rather on the introduction of the evaluation framework, itself. The results of our approach are presented in Figure 3 and its companion table entitled"Three Binary Domains Projection Legend".

The results show that all the methods, except for *SMO* (8) and *NB* (9), fall within the same range. *SMO* and *NB* produce much worse results, since they are further away from *Ideal (1)* than the other approaches; and are shown to behave very differently from one another as well, since they are not clustered together. To better understand the graph, we consider this result in view of the results obtained by the traditional measures of performance that are displayed in Table 1, for the three domains considered.

This comparison tells us something interesting: *SMO* fails quite miserably according to all three measures (Accuracy,

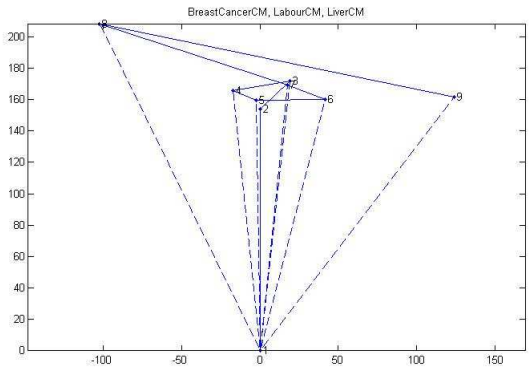

Figure 3: Projection of Three Binary Domains

Three Binary Domains Projection Legend			
Classifier number	Classifier name	Distance from origin	Distance from previous classifier
1	Ideal	0	
2	RandFor	154	
3	Ibk	173	26
4	JRip	167	37
5	Adaboost	160	16
6	Bagging	166	44
7	J48	170	26
8	SMO	232	126
9	NB	203	230

F-measure and AUC) on the `Liver` data set. *NB*, on the other hand, only fails badly on this domain when accuracy is considered. The F-Measure and AUC do not signal the presence of a problem. This means that, unless accuracy was considered, we would not have detected a difference in the behaviour of *NB* on the `Liver` data set. In contrast, our method identified both the problems with *NB* and *SMO* and stated that they were of a different nature. Our method seems to warn us that these two classifiers are sometimes unreliable, whereas the other systems are more stable. Of course, if we had used a different distance measure, the results would have been different. The purpose of our discussion is not so much to compare Euclidean distance to accuracy, F-measure and AUC. Instead, we wish to point out how differences between classifiers are clearly and immediately noticeable from our graph.

Please note that *SMO*'s lower performance on the Liver data is something that would not have been picked up (except possibly if the F-measure had been considered) by an averaging of performance on all domains since *SMO* gets averages of: 72.46% in accuracy, .44 in F-measure and .65 in AUC versus 74.7% accuracy, .64 in F-measure and .75 in AUC, for *Adaboost* (5), quite a good classifier on these domains. Once its performance results averaged, NB would not have exhibited any problem whatsoever, no matter which traditional evaluation method were considered. Indeed, it produced averages of: 72.2% for accuracy, .67 for the F-measure, and .77 for the AUC, three results that are compa-

		Accuracy	F-Measure	AUC
NB	BC:	71.70	0.48	0.70
	La:	89.50	0.92	0.97
	Li:	55.40	0.60	0.64
J48	BC:	77.50	0.40	0.59
	La:	73.70	0.79	0.7
	Li:	68.70	0.59	0.67
Ibk	BC:	72.40	0.41	0.63
	La:	82.50	0.86	0.82
	Li:	62.90	0.56	0.63
JRip	BC:	71	0.43	0.60
	La:	77.20	0.83	0.78
	Li:	64.60	0.53	0.65
SMO	BC:	69.60	0.39	0.59
	La:	89.50	0.92	0.87
	Li:	58.30	0.014	0.50
Bagging	BC:	67.8	.23	.63
	La:	86	0.90	0.88
	Li:	71	0.624	0.73
Adaboost	BC:	70.30	0.46	0.70
	La:	87.70	0.91	0.87
	Li:	66.10	0.534	0.68
RandFor	BC:	69.23	0.39	0.63
	La:	87.70	0.91	0.90
	Li:	69	0.64	0.74

Table 1: Performance by Traditional Measures on the Breast Cancer (BC), Labour (La) and Liver (Li) domains.

rable to those obtained by *AdaBoost*, our reference. Once again, what is remarkable about our visualization approach is that the graph of Figure 3 tells us immediately that an abnormal situation has been detected with respect to *SMO* and *NB* and that this problem is of a different nature in each case. This is quite useful given how tedious and mistake-bound the reading of large result tables can be. Our approach can be used to filter out problem spots, that can then be carefully analyzed, using only the portion of the result tables that focus on this problem spot.

Though we only used binary domains in this example, we could have, instead, mixed binary and multi-class domains using the same approach, thus finding a way to aggregate values that could not, otherwise, be aggregated together.

4 Experiments on Single MultiClass Domains using Confusion Matrices

In this section, we consider how our approach fares on multiclass domains. In particular, we consider the `Anneal` domain from UCI. `Anneal` is a 6-class domain (though one of the classes is represented by no data point). The data set is quite imbalanced since the classes contain 684, 99, 67, 40, 8 and 0 instances, respectively. The results obtained on this domain are displayed in Figure 4 along with the companion table entitled "Anneal Projection Legend". This time, the graph encourages us to beware of *NB* (8) and *Adaboost* (9), though it also shows us that *Adaboost* and *NB*'s problems are not related. We compare the results of Figure 4 to the

NB	J48	Ibk	JRip	SMO	Bag	Boost	RandFor
86.30	98.40	99.10	98.30	97.40	98.20	83.60	99.30

Table 2: Accuracies on the Anneal Data Set

accuracy results obtained on this domain, displayed in Table 2.

While the accuracies (the only simple compact measure that can be used in multi-class domains) suggest that *NB* and *Adaboost* do not classify the data as well as the other domains, it does not alert us of the seriousness of the problem to the same extent that our approach does. Indeed, while it is true that *NB*'s accuracy of 86.3% is comparatively much lower than *SMO*'s accuracy of 97.4%, because in and of itself 86.3% is not a bad accuracy on a 6-class problem, it is conceivable that if a user had a specific interest in using *NB* rather than *SMO* or any other good method, s/he could decide that the tradeoff in accuracy is not worth a switch to a classifier other than *NB* since *NB*'s accuracy is good enough for his/her particular application. This is quite different from the story painted in Figure 4 in which *SMO* and *Adaboost* are exaggeratedly far from the ideal in comparison to the other classifiers.

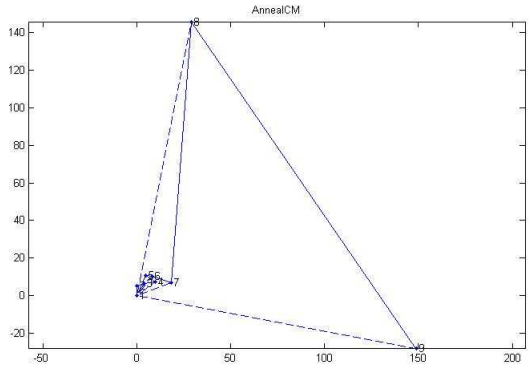

Figure 4: Projection of the results on a MultiClass domain: Anneal

Anneal Projection Legend			
Classifier number	Classifier name	Distance from origin	Distance from previous classifier
1	Ideal	0	
2	RandFor	5	
3	Ibk	7	4
4	J48	12	6
5	JRip	12	6
6	Bagging	13	3
7	SMO	20	11
8	NB	148	139
9	Adaboost	151	211

In order to interpret the results, it is important to remember that the `Anneal` problem is severely imbalanced. The effects of this imbalance are clearly seen in the confusion matrices of *Adaboost* and *NB* in Tables 3 and 4.

Predicted/ True class	a	b	c	d	e	f
a	0	0	8	0	0	0
b	0	0	99	0	0	0
c	0	0	684	0	0	0
d	0	0	0	0	0	0
e	0	0	0	0	67	0
f	0	0	40	0	0	0

Table 3: The confusion Matrix for AdaBoost

Predicted/ True class	a	b	c	d	e	f
a	7	0	1	0	0	0
b	0	99	0	0	0	0
c	3	38	564	0	0	79
d	0	0	0	0	0	0
e	0	0	0	0	67	0
f	0	0	2	0	0	38

Table 4: The confusion Matrix for NB

As shown in Table 3, *Adaboost* only gets the points from the largest class and the third largest class well-classified, ignoring all the other classes. From Table 4, we see that *NB* classifies all the classes accurately, except for the largest class. We do not have enough space, here, to include the confusion matrices of the other methods, but we can report that they all did quite a good job on all classes. In effect this means that all the classifiers but *NB* and *Adaboost* are able to deal with the class imbalance problem, and that *NB* and *Adaboost* both behave badly on this domain, although they do so in different ways. This is exactly what the graph of Figure 4 tells us. The accuracy results do suggest that *NB* and *Adaboost* have problems, but they do not differentiate between the two kind of problems.

5 Multi-facetted Classifier Evaluation

The purpose of this section is to explore the kind of advantages our framework's flexibility can provide. We begin by pointing out that the visualizations we displayed in our previous graphs are only relative assessments. For example, in the graph of Figure 4, we can see that all the classifiers, aside from *NB* (8) and *AdaBoost* (9) are very close together. After viewing the entire graph, we may want to zoom in on the tight cluster formed of classifiers 2 to 7, included. This is done in Figure 5 (whose legend is the same as that of Figure 4).

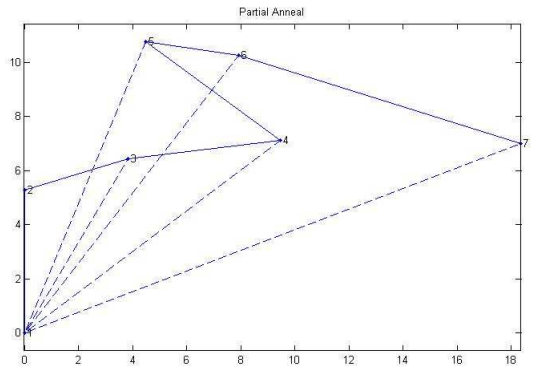

Figure 5: Projection of the partial results on a MultiClass domain: Anneal

From this figure, we can see that *SMO* (7) does not perform as well as the other classifiers (though a lot better than *NB* and *AdaBoost* in Figure 4), that *RandFor* (2) and *IBk* (3) are the best classifiers on this problem, followed by *J48* (4), *JRip* (5) , which are somewhat equivalent in performance (though somewhat different from one another) and, finally, *Bagging* (6). An implementation that would allow the user to zoom in and out of graphs in that fashion would, thus, be quite a useful analytical tool.

Another issue we wish to investigate is the use of different distance measures. All our experiments, thus far used the Euclidean distance (L2 Norm), we wondered what the outcome would be if we were to use the Manhattan distance (L1 Norm), instead. The results are shown in Figure 6 which comes accompanied by the table entitled "Anneal L1 Norm Projection Legend".

There is only one qualitative difference between the graphs produced by the L1 and the L2 norms: *NB* (8) appears closer to ideal than *Adaboost* (9) in Figure 6, than it did in Figure 4. Since the L2 norm penalizes the presence of major concentrated misclassification errors more than the presence of small ones (since each concentration of error gets squared), and the L1 norm simply counts the number of misclassification errors present, we can reason that *NB* makes fewer errors than *Adaboost*, altogether, but that the majority of its errors are concentrated in one or a few large spots. In contrast, we can reason that although *Adaboost* makes more errors than *NB* altogether, its errors are more broadly distributed and appear in large numbers of small clusters. Another look at the confusion matrices of Tables 3 and 4 confirms this hypothesis. Indeed, we see that *Adaboost* makes 147 mistakes versus 123 for *NB*, thus explaining *NB*'s better performance with the L1 norm. In addition, since we see that, inconsiderate of class E, on which the two classifiers behave the same way, *NB* makes its major mistakes on class C, the largest class, whereas *Adaboost* makes no mistake on class C, but, instead, misclassifies all the other, smaller classes (except for class E), we understand where the results obtained with the L2 norm, which equate *Adaboost* and *NB*'s performance, come from. Thus, we can see how, provided that we understand the meaning of the various distance measures we may use, each of them used simultaneously can quickly give us some important insight into the comparative performance of our classifiers.

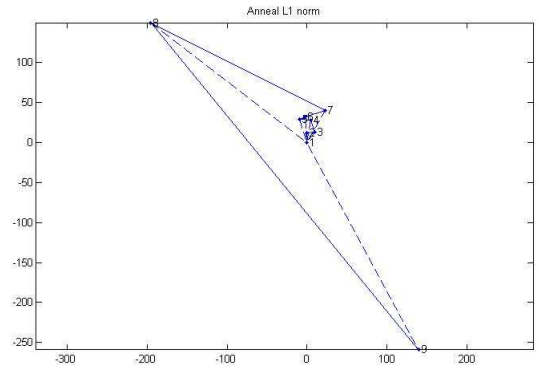

Figure 6: Projection of the results on a MultiClass domain, using the L1 Norm: Anneal

Anneal L1 Norm Projection Legend			
Classifier number	Classifier name	Distance from origin	Distance from previous classifier
1	Ideal	0	
2	RandFor	12	
3	IBk	16	10
4	J48	28	16
5	JRip	30	14
6	Bagging	32	8
7	SMO	46	26
8	NB	246	244
9	AdaBoost	294	528

In the last part of this study, we consider other views of our `Anneal` domain: L2 norm views in which the multiclass problem has been reduced to two classes. In particular, because of space constraints, we focus on the view of Class A versus all the other classes.

Such views also provide interesting insight into the behaviour of our classifiers. For example, in Figure 7, which

Anneal Class A Projection Legend			
Classifier number	Classifier name	Distance from origin	Distance from previous classifier
1	Ideal	0	
2	SMO	2	
3	RandFor	2	0
4	IBk	2	0
5	Bagging	5	4
6	J48	5	0
7	JRip	7	2
8	Adaboost	14	7
9	NB	4	14

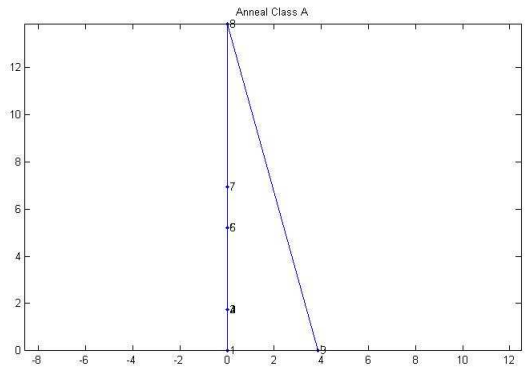

Figure 7: Projection of the results on a MultiClass domain, compressed into two classes: Anneal, Class A

comes accompanied by the table "Anneal Class A Projection Legend", we learn that *NB* (9) is not a bad classifier with respect to class A. Rather, we can see that *JRip* (7) is the one that causes problems on that class, along with *AdaBoost* (14), which remains the most problematic of them all. Looking at Tables 3 and 4 , as well as at the confusion matrix for JRip (which cannot be included here for lack of space), confirms the relative positions of *Adaboost*, *NB* and *JRip* in Figure 7. Indeed, *Adaboost* misclassifies all 8 examples of class A, but does not make any false positive errors with respect to A; *NB* only misclassifies 1 example of class A and makes 3 false positive errors on it; and *JRip* misclassifies 4 class A examples and does not make any false positive errors on it. Note that if the L1 norm were used instead of the L2 norm, *JRip* and *NB* would each be as far away from ideal as the other. Because the L2 norm emphasizes the fact that *JRip* (and *Adaboost*, even more) have a high concentration of errors in a single spot, these two classifiers fare worse than *NB* on this problem.

We conclude that our framework allows us to pinpoint tradeoffs between the different measures and focuses we choose quite rapidly. These are not as clear when using traditional evaluation methods, which are not inherently visual. Because human beings tend to process visual information much faster and probably better than they do other kinds of information—"A picture is worth a thousand words"—we suggest that our approach has great potential for the future.

6 Conclusion and Future Work

We conclude this study by summarizing our findings and suggesting areas for future work.

6.1 Summary

This paper presented a new evaluation method which, rather than aggregating the entries of the confusion matrices pertaining to the performance of a classifier into a single measure, treats all the performance data pertaining to that classifier as a high-dimensional vector. The vectors representing classifiers are then projected into a 2-dimensional space by a distance-preserving projection method. This approach presents several advantages, including the fact that it offers a visualization method that allows data mining practitioners to spot immediately any irregularity in the behaviour of their classifiers. It also indicates whether the detected irregularities are similar to each other or not. This particular method is, but one implementation of the general framework we advocate that views the problem of classifier evaluation as one of analyzing high-dimensional data.

6.2 Future Work

As presented, our approach may appear limited to the comparison of single classifier's performance, thus precluding the evaluation of threshold-insensitive classifiers and the computation of statistical guarantees in our results. Indeed, unlike ROC Analysis, (Fawcett 2003) and Cost-Curves (Drummond & Holte 2006), our current approach is threshold-sensitive. This restricts its use to balanced data sets with known cost-matrices (if the cost-matrix is known, it can be integrated to our distance function). In case where the costs are unknown, it cannot be used as presented here. The advantages of our method over ROC Analysis, and Cost-Curves, however, are the same as those described earlier: ROC Analysis and Cost-Curves uses biased summaries of the confusion matrices in their computations and are applicable to single binary problems only.

We believe that we could use our framework to analyze classifiers in a threshold insensitive way as well. This way, we would be coupling the advantages of ROC Analysis and Cost-Curves to those of our current method. We could, for example, concatenate the results obtained by the same classifier using different thresholds within a single vector and project that vector into our two-dimensional space. Alternatively, we could separate the performance at each threshold, projecting a single point for each classifier at each threshold level. This would create clouds of classifier performance that could then convey the same kind of information (though more detailed) than ROC graphs.

With respect to the computation of statistical guarantees, we believe that we could easily integrate the results of several cross-validation folds within a single vector or plot the results of each fold for each classifier, giving us, as above a cloud of points for each classifier that would, this time, offer a visualization of the variance of that classifier. More formally, we could then apply a statistical test to the results of this projection.

We are also planning to expand our understanding of our framework by experimenting more thoroughly with different performance data representations (e.g., the outcome of classification on each test data point), different projection methods, as well as different distance measures. We believe that once it is carefully studied, this framework could become an integral part of the classifier evaluation process.

References

C.Blake, and Merz, C. 1998. Uci repository of machine learning databases.

Drummond, C., and Holte, R. 2006. Cost curves: An im-

proved method for visualizing classifier performance. *Machine Learning* 65(1):95–130.

Fawcett, T. 2003. ROC graphs: Notes and practical considerations for data mining re searchers. Technical Report HPL-2003-4, HP Laboratories, Palo Alto, CA.

Jain, A. K.; Duin, R. P.; and Mao, J. 2000. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(1).

Lee, R.; Slagle, J.; and Blum, H. 1977. A Triangulation Method for the Sequential Mapping of Points from N-Space to Two-Space. *IEEE Transactions on Computers* 26(3):288–292.

Witten, I. H., and Frank, E. *WEKA Software, v3.5.2*. University of Waikato.

Yang, L. 2004. Distance-preserving projection of high dimensional data. *Pattern Recognition Letters* 25(2):259–266.