

# Why Question Machine Learning Evaluation Methods? (An illustrative review of the shortcomings of current methods)

Nathalie Japkowicz

School of Information Technology and Engineering  
University of Ottawa  
800 King Edward Avenue  
Ottawa, Ontario, K1N 6N5  
nat@site.uottawa.ca

## Abstract

The evaluation of classifiers or learning algorithms is not a topic that has, generally, been given much thought in the fields of Machine Learning and Data Mining. More often than not, common off-the-shelf metrics such as Accuracy, Precision/Recall and ROC Analysis as well as confidence estimation methods, such as the t-test, are applied without much attention being paid to their meaning. The purpose of this paper is to give the reader an intuitive idea of what could go wrong with our commonly used evaluation methods. In particular, we show, through examples, that since evaluation metrics and confidence estimation methods *summarize* the system's performance, they can, at times, obscure important behaviors of the hypotheses or algorithms under consideration. We hope that this very simple review of some of the problems surrounding evaluation will sensitize Machine Learning and Data Mining researchers to the issue and encourage us to think twice, prior to selecting and applying an evaluation method.

## Introduction

The purpose of evaluation in Machine Learning is to determine the usefulness of our learned classifiers (hypotheses) or of our learning algorithms on various collections of data sets. Optimally, the evaluation process would include the following careful steps, as suggested by [Elazmeh, 2006]:

1. Decide what “interesting” properties of the classifier should be measured, and choose an evaluation metric accordingly.
2. Decide what confidence estimation method should be used to validate our results.
3. Check that the assumptions made by the evaluation metric and the confidence estimation method are verified on the domain under consideration.
4. Run the evaluation method, using the chosen metric and confidence estimation method, and analyze its results.
5. Interpret these results with respect to the domain.

Unfortunately, many Machine Learning/Data Mining researchers consider step 1 of this framework very lightly, and skip steps 2, 3 and 5, to concentrate, mainly, on step 4. In this paper, we only focus on the problems

caused by neglecting Steps 1, 2 and 3. Our considerations on Machine Learning Evaluation will be divided into two parts: those regarding

- The performance metrics employed
- The confidence estimation method chosen

The first part will consider what happens when bad choices of performance metrics are made (Step 1 is considered too lightly). The second part will look at what happens when the assumptions made about the confidence estimation method chosen are not respected (Steps 2 and 3 are disregarded).

More specifically, the purpose of this paper is to review the evaluation methods—metrics and confidence estimation—commonly used in the field of Machine Learning/Data Mining and review, through examples, how these methods are lacking.

It is worth noting that the approach to evaluation in Machine Learning<sup>1</sup> is quite different from the way in which evaluation methods are approached in most applied fields such as Economics, Psychology, Biology, and so on. Such a disparity can be explained by the fact that, in Machine Learning, more weight is given to the creation of sophisticated algorithms for solving problems than to the observations that lead to this algorithm or its testing, whereas research in the other fields mentioned applies greater reliance on thorough testing of existing theories, prior to suggesting new ones. Whatever the reason for this state of affairs, we must recognize that the Machine Learning standing on the issue of evaluation is weak, and results in the creation of a multitude of learning methods that never get thoroughly assessed. In particular these methods may get undue credit, or, alternatively, not get sufficiently recognized. This has practical ramifications as well, as colleagues in areas that could benefit from our approaches may dismiss (some, actually have!) our methods because of lack of guarantees on them. By bringing our evaluation strategies to the

---

<sup>1</sup> In the remainder of the paper, the term “Machine Learning” covers both the areas of Machine Learning and Data Mining.

True class →	Positive	Negative
<b>Hypothesized Class</b>		
<b>Yes</b>	<b>TP</b>	<b>FP</b>
<b>No</b>	<b>FN</b>	<b>TN</b>
	<b>P= TP+FN</b>	<b>N = FP+TN</b>

Table 1: A confusion matrix,  $TP$ = True Positive count,  $FN$ =False Negative count,  $FP$ =False Positive count, and  $TN$ =True Negative count

standards by which practitioners in other fields judge their methods' performance, we hope to open up our field to their scrutiny and favor greater cross-discipline exchanges.

Let us now turn to the problems that can arise during the evaluation phase of our learning systems if Steps 1, 2 and 3 of our above framework have been neglected. Once a learning algorithm is run, it is worth noting that many quantities can be computed from the results it obtained. Since the purpose of evaluation is to offer simple and convenient ways to judge the performance of a learning system or a hypothesis and/or to compare it to others, evaluation methods can be seen as *summaries* of the systems' performance. Furthermore, the same data can be summarized in various ways, and, sometimes, with different assumptions made on it, giving rise to different performance evaluation metrics or confidence evaluation tests. We will show how three of the most commonly used performance evaluation metrics—Accuracy, Precision/Recall, ROC Analysis—do focus on various aspects of a classifier, but, at the same time, ignore others, thus grouping together, by assigning them the same performance values, classifiers that exhibit extremely different behaviors. We will also show how, by the same process of summarization (and violated assumptions), the most commonly used statistical confidence estimation method—the t-test—fails to differentiate learning systems exhibiting, again, very different behaviours. Though we do not advocate dropping the metrics and confidence estimation methods in use, we do suggest being aware of their restrictions.

The remainder of the paper is divided into three sections. Section 2 focuses on the issue of performance metrics. More specifically, it demonstrates, through a number of examples the shortcomings of Accuracy, Precision/Recall and ROC Analysis. Section 3 looks at the issue of confidence estimation and, again, demonstrates, through an example the shortcomings of the confidence estimation method commonly used: the t-test, applied to a 10-fold cross-validation process. Section 4, finally, discusses directions for further research into these issues.

### Shortcomings of our Performance Metrics

In this section, we consider the three most commonly used metrics in Machine Learning: Accuracy (or

equivalently, Error Rate), Precision/Recall and ROC Analysis. In each case, we begin by stating the advantages of these methods, showing how each answers issues left unanswered by the previous ones, but we, then, continue by explaining the shortcomings they each have.

All our discussions are based on the definition of a confusion matrix. The confusion matrix we use is depicted in table 1. The definitions of the measures under consideration in this paper are given below, based on this table.

- Accuracy =  $(TP + TN) / (P + N)$
- Precision =  $TP / (TP + FP)$
- Recall =  $TP / P$
- FP Rate =  $FP / N$

**Note:** ROC Analysis uses the Recall and FP Rate measures as will be explained below.

### What is wrong with Accuracy?

Before discussing what is wrong with Accuracy, let us say a few words about what is right about it. Accuracy is the simplest, most intuitive evaluation measure for classifiers: since we are interested in creating classifiers that do not make any mistakes, what is simpler than counting the number of mistakes that they make and comparing them based on this measure? While this represents a simple solution, widely used in our field, it is worth noting that Accuracy does not distinguish between the types of errors it makes (False Positive versus False Negatives). While this is acceptable if the data set contains as many examples of both classes (i.e., if it is a balanced data set) and if accuracy is very high, it is difficult to imagine an application which, in any other cases, would not need to distinguish between the two kinds of errors. As a matter of fact, in certain circumstance, ignoring the issue can lead to catastrophic results like in the case of a medical classification problem whose goal is to discriminate cancerous (positive class) from non cancerous patients (negative class).

We illustrate the problem more specifically with the following numerical example: Consider two

classifiers represented by the two confusion matrices of Table 2. These two classifiers behave quite differently. The one symbolized by the confusion matrix on top does not classify positive examples very well, getting only 200 out of 500 right. On the other hand, it does not do a terrible job on the negative data, getting 400 out of 500 well classified. The classifier represented by the confusion matrix on the bottom does the exact opposite, classifying the positive class better than the negative class with 400 out of 500 versus 200 out of 500. It is clear that these classifiers exhibit quite different strengths and weaknesses and shouldn't be used blindly on a data set such as in the medical domain we previously described. Yet, both classifiers exhibit the same accuracy of 70%. Precision and Recall do address this issue but present another disadvantage that is discussed below.

True class → Hypothesized class	Positive	Negative
Yes	200	100
No	300	400
	P = 500	N = 500

True class → Hypothesized class	Positive	Negative
Yes	400	300
No	100	200
	P = 500	N = 500

Table 2: The trouble with Accuracy: Two confusion matrices yielding the same accuracy despite serious differences

### What is wrong with Precision/Recall?

Once again, let us start with what is right with Precision and Recall. Although these measures are not quite as simple and intuitive as accuracy is, they still have a relatively straightforward interpretation. Precision assesses to what extent the classifier was correct in classifying examples as positives, while Recall assesses to what extent all the examples that needed to be classified as positive were so.

As mentioned before, Precision and Recall have the advantage of not falling into the problem encountered by Accuracy. Indeed, considering, again, the two confusion matrices of Table 2, we can compute the values for Precision and Recall and obtain the following results:

- Precision = 66.7% and Recall = 40% in the first case, and
- Precision = 57.2% and Recall = 80% in the second

These results, indeed, reflect the strength of the second classifier on the positive data, with respect to the first classifier. This is a great advantage over accuracy.

What Precision and Recall do not do, however—which, incidentally, accuracy does—is place any judgment on how well a classifier decides that a negative example is, indeed, negative. It is possible, for example, for a classifier trained on our medical domain to have respectable precision and recall values even if it does very poorly at recognizing that a truly healthy patient is, indeed, healthy. This is disturbing since the same values of precision and recalls can be obtained no matter what the *proportion* of patients *labeled* as healthy truly *is* healthy.

True class → Hypothesized Class	Positive	Negative
Yes	200	100
No	300	400
	P = 500	N = 500

True class → Hypothesized Class	Positive	Negative
Yes	200	100
No	300	0
	P = 500	N = 100

Table 3: The trouble with Precision and Recall: Two confusion matrices with the same values of precision and recall, but very different behaviors

More specifically, consider, as an extreme situation, the confusion matrices of table 3.<sup>2</sup> The matrix on top is the same as the top matrix of Table 2, whereas the one on the bottom represents a new classifier tested on a different data set. Although both classifiers have the same Precision and Recall of 66.7% and 40%, respectively, it is clear that the classifier represented by the confusion matrix on the bottom presents a much more severe shortcoming than the one on top since it is incapable of classifying true negative examples as negative (it can, however, wrongly classify positive examples as negative!). Such a behavior, by the way,

<sup>2</sup> It can be argued that the problem illustrated by this example would not occur, in practical situations, since the classifiers would be evaluated on similarly distributed data sets. However, our purpose, here, is not to compare classifiers as much as it is to challenge the *meaning* of our metrics. We, thus, deem it acceptable to use matrices representing different data distributions to illustrate our point.

as mentioned before is reflected by the accuracy measure which assesses that the classifier on the bottom is only accurate in 50% of the cases while the classifier on the top is accurate in 70% of the cases. This suggests that Precision and Recall are quite blind, in a certain respect, and might be more useful when combined with accuracy or when applied to both the positive *and* the negative class. The argument against combining these three measures, however, is that evaluation methods are supposed to *summarize* the performance. If we must report the values of three measures, then why not simply return the confusion matrices, altogether?

The next section turns to ROC Analysis which corrects for both the problems of Accuracy, and Recall and Precision, but introduces, yet, a new problem.

### What is wrong with ROC Analysis?

As before, prior to discussing the negative aspects of ROC Analysis, we will focus on its positive ones. ROC Analysis is, once again, a method that is not as simple and straightforward as accuracy, but which, nonetheless, still has intuitive appeal. It should be thought of in two respects:

- The measure it uses
- The manipulation it does of these measures

To begin with, the measures it uses are the Recall (TP/P) and the False Positive Rate (FP/N). ROC Analysis plots the FP Rate on the x-axis of a graph and the Recall on the y-axis. The points on the graph can be interpreted as reflective of a greater performance if they are simultaneously located close to 0 on the x-axis and close to 1 (or 100%) on the y-axis. This is because the closer a point is to 0 on the x-axis, the smaller its false-positive rate is; and the closer to 1 a point is on the y-axis, the higher its recall.

With these measures established, ROC Analysis then considers a classifier as made of two parts. The first one (the hard part, which also defines the classifier) performs a kind of data transformation bringing similar examples of the same class closer to each other. The second part (much easier to handle) decides where exactly, the boundaries between the different clusters established in the first part should be placed. The algorithm is then tested on various values of the FP rate by allowing the boundaries to move from locations representing a very small to locations representing a very large FP rate. This allows an ROC graph to view at what costs (FP-Rate, i.e., False Alarms) benefits (Recall, i.e., also Hit Rate) can be achieved. For a more thorough description of ROC Analysis, please, see [Fawcett, 2003]. Also see the discussion in [Drummond, & Holte, 2005] for a more detailed discussion of the tradeoff between false alarms and hit rate.

True class → Hypothesized class	Positive	Negative
Yes	200	10
No	300	4,000
	P = 500	N = 4,010

True class → Hypothesized Class	Positive	Negative
Yes	500	1,000
No	300	400,000
	P = 800	N = 401,000

Table 4: Two confusion matrices representing the same point in ROC space, but with very different precisions

In this discussion, we only consider the first aspect of ROC Analysis: the performance measure it uses. The great advantage of this performance measure is that it separates the algorithm's performance over the positive class from its performance over the negative class. As a result, it does not suffer from either of the two problems we considered before. Indeed, in the case of Table 2, the top classifier is represented by ROC graph point (0.2, 0.4) while the bottom classifier is represented by point (0.6, 0.8). This clearly shows the tradeoff between the two approaches: although the top classifier makes fewer errors on the negative class than the bottom one, the bottom one achieves much greater performance on the positive class than the top one. In the case of Table 3, the top classifier is the same classifier as the top classifier of Table 2 and is thus represented by the same point, (0.2, 0.4) whereas the bottom classifier is represented by point (1, 0.4). As should be the case, the second classifier appears as problematic right away, having the largest possible FP rate, yet achieving the same recall as the top classifier which reports only 1/5<sup>th</sup> of the FP rate of the bottom one. It is, thus clear that ROC Analysis has great advantages over Accuracy, Precision and Recall.

Nonetheless, there are reasons why ROC analysis is not an end in itself, either. For example, ROC Analysis can give a high rating to a classifier that **classifies** as *cancerous*, a high proportion of patients that *are healthy*<sup>3</sup>. This is illustrated numerically in the following example. Consider the two confusion matrices of Table 4.<sup>4</sup> The classifier represented by the confusion matrix on the bottom generates a point in

<sup>3</sup> Please, note the contrast with Precision/Recall (P/R). In sum, neither P/R nor ROC pays much attention to the true negative class, but they disregard it in different ways.

<sup>4</sup> Please, see footnote 2.

ROC space that is on the same vertical line as the point generated by the classifier represented by the confusion matrix on top ( $x = \text{FP rate} = 0.25\%$ ), but that is substantially higher (by 22.25%, [ $\text{recall}_{\text{top}} = 40\%$ ;  $\text{recall}_{\text{bottom}} = 62.5\%$ ]) than the one on top. This suggests that the classifier on the bottom is a better choice than the one on the top (at least for this point in ROC space), yet, when viewed in terms of precision, we see that the classifier on top is much more precise, with a precision of 95.24% than the one on the bottom, with a precision of 33.3%. Ironically, this problem is caused by the fact that ROC Analysis nicely separates the performance of the two classes, thus staying away from the previous problems encountered by Accuracy and Precision/Recall.

### Shortcomings in Confidence Estimation

So far, we have focused on the issue regarding the choice of an evaluation metric. We have shown that no metric answers all the questions and choosing a metric blindly is not a good strategy. We now turn to the question of selecting an appropriate confidence estimation method. We place ourselves in the context in which a learning algorithm is trained and tested on a data set using 10-fold cross-validation together with accuracy. 10-fold cross-validation consists of dividing the data set into 10 non-overlapping subsets (folds) of equal size and running 10 series of training/testing experiments, combining 9 of the subsets into a single training set and using the last subset as a testing set. The experiment is repeated, rotating the identity of the testing set. [See [Mitchell, 1997] for more detail]. Along with returning the mean obtained on classifying the various folds, it has become common, in Machine Learning, to indicate the confidence interval in which these results reside. In the next section, we will see how, because the t-test *summarizes* the results obtained on each fold, important information regarding the behavior of the learning algorithm is lost in the process of making this calculation.

### What is wrong with the t-test?

Assume that we have two classifiers: one that always performs approximately the same way, in terms of the metric used, and the other, that performs much more erratically, often doing very well, but sometimes doing quite poorly. It is, then, possible for the t-test to return the same confidence value for both classifiers. This could cause practical problems since the t-test encourages us to trust the erratic classifier to the same extent as to which we trust the more stable one, even though, there could be a non-insignificant probability that on some case, the erratic classifier would perform in quite a substandard way. This, obviously, is a problem in highly sensitive tasks such as the discrimination between cancerous and

non-cancerous patients. We now illustrate this issue numerically.

Let us use the paired t-test confidence estimation approach in the following setting. We are given a Procedure G which performs according to a given gold-standard. We want to compare two learning algorithms A1 and A2 to G. We are told that an algorithm that always performs within 8% of G is acceptable. We use the procedure described in [Mitchell, 97; p.147] to perform 10-fold cross-validation and compute a confidence interval based on the paired t-test. We assume that our performance measure obtains the results listed in Table 5, where the  $\delta_i$  represent the differences in accuracy between A1 and G (case 1) and A2 and G (case 2).

From the table, we can see that the average difference in accuracy is  $\delta = 0\%$  in both cases. Furthermore, as described by [Mitchell, 1997], the confidence interval for these differences is:

$$\delta \pm t_{N,9} s_{\delta}$$

where  $s_{\delta} = \sqrt{1/90 \sum_{i=1}^{10} (\delta_i - \delta)^2}$ , and  $t_{N,9}$  is a constant, chosen depending on the desired confidence level. Substituting for the  $\delta_i$  's and  $\delta$ , we get  $s_{\delta} = \sqrt{1/90 * 250}$  in both cases, meaning that the differences of the two algorithms with respect to G have the same mean and the same confidence interval around this mean. Yet, it is clear that the two algorithms behave quite differently, since A1 is a very stable procedure, whereas A2 is not (something that should be known by the user). As a matter of fact, given our constraint on performance within 8% of G, it turns out that A1 is acceptable whereas A2 is not, a piece of information not relayed to us by the confidence interval.

This, once again, suggests a serious flaw in the way in which we use one of our best trusted evaluation tool. The flaw comes from the fact that the t-test makes the assumption that the population to which it is applied is *normally distributed*. Obviously the second line in Table 5 is not distributed normally, and the t-test, should actually not have been applied to this case. It is an instance where Step 3 of the framework proposed by [Elazmeh, 2006] was skipped.

### Future Directions

Altogether, this paper has argued that choosing performance metrics and confidence estimation methods blindly and applying them without any regard for their meaning and the conditions governing them, is not a particularly interesting endeavor and can result in dangerously misleading conclusions. Instead, the issue of evaluation should be re-considered carefully, every time a new series of tests is planned. Please, note

that, we are not the first authors to have written on this topic. Most notable are the papers by [Salzberg, 1997], [Dietterich, 1998], [Provost & Fawcett, 2001].

There are many future directions that, we hope, this paper will suggest. Here are some that we want to follow in the short term. First, we are planning to formalize this paper. For the time being, we only explained, through a small number of examples, aspects of current evaluation methods in Machine Learning that require further thought than is currently given to them. We are planning to formalize this discussion using the framework of [Elazmeh, 2006]. We also hope to expand our discussion to other evaluation metrics, such as the Lift, Break-Even Point, etc...and other confidence estimation methods, such as

	<b>Fold1</b>	<b>Fold2</b>	<b>Fold3</b>	<b>Fold4</b>	<b>Fold5</b>	<b>Fold6</b>	<b>Fold7</b>	<b>Fold8</b>	<b>Fold9</b>	<b>Fold10</b>
	<b><math>\delta_1</math></b>	<b><math>\delta_2</math></b>	<b><math>\delta_3</math></b>	<b><math>\delta_4</math></b>	<b><math>\delta_5</math></b>	<b><math>\delta_6</math></b>	<b><math>\delta_7</math></b>	<b><math>\delta_8</math></b>	<b><math>\delta_9</math></b>	<b><math>\delta_{10}</math></b>
Case1	+5%	-5%	+5%	-5%	+5%	-5%	+5%	-5%	+5%	-5%
Case2	+10%	-5%	-5%	0%	0%	0%	0%	0%	0%	0%

Table 5: Difference in accuracy of A1 versus G (case 1) and A2 versus G (case 2) for each fold of the cross-validation procedure

## Acknowledgements

I would like to thank William Elazmeh and Chris Drummond for their thorough comments on a previous draft of this paper, as well as the workshop reviewers. This research is supported by Canada's National Science and Engineering Research Council.

## References

[Dietterich, T.G., 1996], *Proper Statistical Tests for Comparing Supervised Classification Learning Algorithms* (Technical Report). Department of Computer Science, Oregon State University, Corvallis, OR.

[Drummond, C. and Holte, R.], 2000, "Explicitly Representing Expected Cost: An Alternative to ROC Representation". *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 198-207.

[Drummond, C. and Holte, R., 2005], "Learning to Live with False Alarms", *Data Mining Methods for Anomaly Detection workshop at the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2005)*. pp. 21-24.

[Elazmeh, 2006], Personal Communication.

the McNemar test or bootstrapping. Through this formalized study, we hope to derive a framework that would link our various evaluation tools to the different types of problems they address and those that they fail to address.

In addition to creating the above-mentioned framework, we hope to pinpoint areas in this Machine Learning evaluation landscape that have not yet been well covered by the current measures and create (or borrow from the literature in other applied fields) new measures for use by the community. We have already started this process with respect to evaluation metrics (see [Sokolova, 2006] in this volume) and confidence estimation methods ([Elazmeh, 2006] in this volume).

[Elazmeh, W., Japkowicz, N., and Matwin, S., 2006], "A Framework for Measuring Classification Difference with Imbalance", *AAAI'2006 Workshop on Machine Learning Evaluation*

[Fawcett, T., 2003], *ROC Graphs: Notes and Practical Considerations for Researchers*, HP Labs Tech Report HPL-2003-4.

[Mitchell, T., 1997], *Machine Learning*, McGraw-Hill.

[Provost, F. & Fawcett, T.], "Robust Classification for Imprecise Environments", *Machine Learning Journal*, vol 42, no.3, 2001

[Salzberg, S.], On Comparing Classifiers: "Pitfalls to Avoid and a Recommended Approach", *Data Mining and Knowledge Discovery* 1:3 (1997), 317-327.

[Sokolova, M., Japkowicz, N. and Szpakowicz, S.], "On Evaluation of Learning from new Types of Data", Submitted to the *AAAI '2006 Workshop on Machine Learning Evaluation*