Classifier Evaluation under Limited Resources

Reuven Arbel RUBISHAG@ZAHAV.NET.IL
Department of Industrial Engineering
Tel-Aviv University, Israel

Lior Rokach LIORRK@BGU.AC.IL
Department of Information Systems Engineering,
Ben-Gurion University of the Negev

Abstract

Existing evaluations measures are insufficient when probabilistic classifiers are used for choosing objects to be included in a limited quota. This paper reviews performance measures that suit probabilistic classification and introduce two novel performance measures that can be used effectively for this task. It then investigates when to use each of the measures and what purpose each one of them serves. The use of these measures is demonstrated on a real life dataset obtained from the human resource field and is validated on set of benchmark datasets.

1. Introduction and Motivation

The aim of classification is to build a classifier (also known as a classification model) by induction from a pre-classified dataset. The classifier can be then used to classify unlabeled objects. Given the long history and recent growth of the field, it is not surprising that several mature approaches to induction are now available to the practitioner.

This paper focuses on applications in which there is a limited quota and a list of new, unlabeled objects, and the decision maker is using a probabilistic classifier to fill in the quota with the objects most likely to achieve "success".

The limited quota is a common situation in real-life applications. Usually organizations have resource limitations that require cost-benefit considerations. Resource limitations prevent the organization from choosing all the instances. For example, in direct marketing applications (Levin and Zahavi, 2005), instead of mailing everybody on the list, the marketing efforts must target the mailing audience with the highest probability to positively respond to the marketing offer without exceeding the marketing budget.

Another example deals with a security officer in an air terminal. Following September 11, the security officer needs to search on all the passengers that are likely to carry dangerous instruments (such as scissors, penknives and shaving blades). For this purpose the officer is using a classifier that is capable to classify each passenger either as class A, which means, "Carry dangerous instruments" or as class B, "Safe". Suppose that searching a passenger is a time consuming task, and that the security officer is capable to check only 20 passengers before each flight. If the classifier has labeled exactly 20 passengers as class A then the officer will check all these passengers. However if the classifier has labeled more than 20 passengers as class A, then the officer is required to decide which predicated class A passenger should be ignored. On the other hand, if less than 20 people were classified as A, the officer, who must work constantly, has to decide who to check from those classified as B after concluding with the class A passengers.

There also cases in which a quota limitation is known to be exist, but the size of the quota in not known in advance. Still the decision maker would like to evaluate the expected performance of the classifier. This case, for example, happens in some countries regarding the number of undergraduate students that can be accepted to a certain department in a state university. The actual quota for a given year is set according to different parameters including governmental budget. In this case the decision maker would like to evaluate several classifiers for selecting the applicants while not knowing the actual quota size. Finding the most appropriate classifier in-advance is important because the chosen classifier can dictate what the important attributes are, i.e. (the information that the applicant should provide the registration and admission unit).

The aim of this paper is to examine various measures for evaluating the performance of probabilistic classifier with limited quota restriction. We begin by reviewing existing measures and then suggest new measures. Finally we examine these measures on a real-world case study and on a various datasets obtained from UCI repository.

2. Evaluation Measures

Classifiers are evaluated based on some "goodness-of-fit" measures which assess how good the model fits the data. To the purpose of this paper we divide these measures into three categories:

1. Measures for evaluating classifiers that are used on unlimited quota.
2. Measures for evaluating classifiers that are used on limited and known in advance quota.
3. Measures for evaluating classifiers that are used on limited and unknown in advance quota.

The following subsections present measures for each category.

2.1 Measures for evaluating unlimited quota

The most common and straightforward approach to evaluate the performance of the classifier is to use a test set of unseen instances that were not used during the training phase. For every instance in the test set, we compare the actual class to the class that was assigned by the trained classifier. A positive (negative) example that is correctly classified by the classifier is called a true positive (true negative); a positive (negative) example that is incorrectly classified is called a false negative (false positive). These numbers can be organized in a confusion matrix shown in Table 1. Based on the values in Table 1, one can define the following measures:

Accuracy is: *(a+d)/(a+b+c+d)*
Misclassification rate is: *(b+c)/(a+b+c+d)*
Precision is: *d/(b+d)*
True positive rate (Recall) is: *d/(c+d)* (1)
False positive rate is: *b/(a+b)*
True negative rate is: *a/(a+b)*
False negative rate is: *c/(c+d)*

Table 1:A confusion matrix

	Predicted negative	Predicted positive
Negative Examples	A	B
Positive Examples	C	D

Accuracy and its complement measure (misclassification rate) are the most common measures for evaluating classifiers. Nevertheless accuracy is not a sufficient measure to evaluate a model with an imbalanced distribution of the class. In such cases, where the data set contains significantly more majority class instances, than minority class instances, one can always select the majority class and obtain good accuracy performance. Well-known performance measures in this case are *precision* and *recall*. Precision measures how many examples classified as "positive" class are indeed "positive". Recall measures how many examples of "positive" class are correctly classified. The above measures are useful to evaluate crisp classifiers that are used to classify an entire dataset.

However in the case discussed here we are interested in selecting the most appropriate instances to fill in the quota. Thus, other measures should be used to evaluate the classifiers.

2.2 Measures for evaluating limited and known in advance quota

2.2.1 Extended precision and recall measures

In probabilistic classifiers, the abovementioned definitions of precision and recall can be extended and defined as a function of a probability threshold t . In this paper we evaluate a classifier based on a given a test set. This test set consists of *n* instances denoted as $(< \boldsymbol{x_1}, y_1 >,...,< \boldsymbol{x_n}, y_n >)$ such that $\boldsymbol{x_i}$ represents the input features vector of instance *i* and y_i represents its true class ("positive" or "negative").

$$\text{Precision}(t) = \frac{\left|\{< \boldsymbol{x}_i, y_i > \ : \hat{P}_M("positive"|\boldsymbol{x}_i) > t \ \text{ and } y_i = "positive"\}\right|}{\left|\{< \boldsymbol{x}_i, y_i > \ : \hat{P}_M("positive"|\boldsymbol{x}_i) > t\right|} \quad (2)$$

$$\text{Recall}(t) = \frac{\left|\{< \boldsymbol{x}_i, y_i > \ : \hat{P}_M("positive"|\boldsymbol{x}_i) > t \ \text{ and } y_i = "positive"\}\right|}{\left|\{< \boldsymbol{x}_i, y_i > \ : \ y_i = "positive"\}\right|} \quad (3)$$

where M represents a probabilistic classifier that is used to estimate the conditional likelihood of an observation x_i to "positive" which is denoted as $\hat{P}_M("positive"|\boldsymbol{x}_i)$. Note the addition of the “hat” - ^ - to the conditional probability estimation is used for distinguishing it form the actual conditional probability. The typical threshold value of 0.5 means the predicted probability of "positive" must be higher than 0.5 for the instance to be predicted as "positive". By changing the value of t , one can control the number of instances that are classified as "positive". Thus, the t value can be tuned to the required quota size. Nevertheless because there might be several instances with the same conditional probability, the quota size is not necessarily incremented by one.

The discussion in this paper is based on the assumption that the classification problem is binary. In case there are more than two classes, adaptation could be easily made by comparing one class to all the others.

2.2.2 ROC Curves

Another measure is the ROC (Receiver operating Characteristic) curves which illustrate the tradeoff between true positive to false positive rates (see for instance Provost *et al.* 1998). Figure 1 illustrates a ROC curve in which the X-axis represents False positive rate and the Y-axis represents True positive rate. The ideal point on the ROC curve would be (0,100), that is all positive examples are classified correctly and no negative examples are misclassified as positive.

The ROC convex hull can also be used as a robust method of identifying potentially optimal classifiers (Provost and Fawcett, 2001). Given a family of ROC curves, the ROC convex hull can include points that are more towards the north-west frontier of the ROC space. If a line passes through a point on the convex hull, then there is no other line with the same slope passing through another point with a larger true positive (TP) intercept. Thus, the classifier at that point is optimal under any distribution assumptions in tandem with that slope (Provost and Fawcett, 2001).

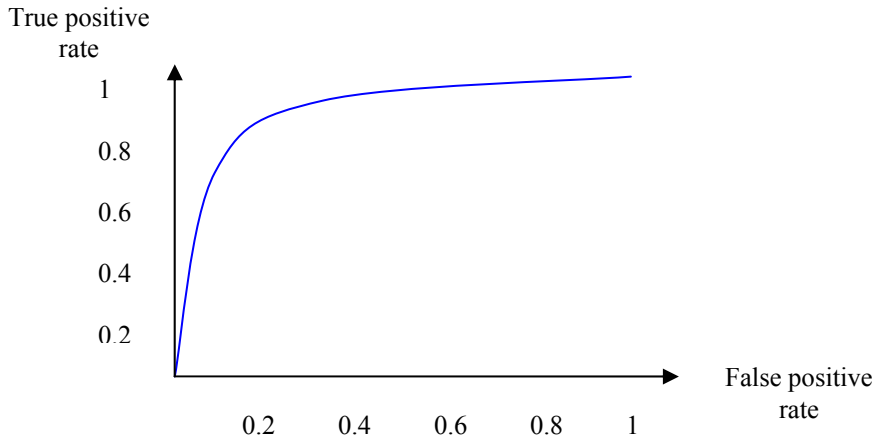

Figure 1: A Typical ROC curve.

2.2.3 Hit Rate Curve

An and Wang (2001) suggested the *hit rate curve*. This curve presents the hit-ratio as a function of the quota size. *Hit-rate* is calculated by counting the actual positive labeled instances inside a determined quota. More precisely for a quota of size j and a ranked set of instances, *Hit-rate* is defined as:

$$\text{HitRate}(j) = \frac{\sum_{k=1}^{j} t^{[k]}}{j} \tag{4}$$

where $t^{[k]}$ represents the truly expected outcome of the instance located in the k'th position when the instances are are sorted according to their conditional probability for "positive" by descending order. Note that if the k'th position can be uniquely defined (i.e. there is exactly one instance that can be located in this position) then $t^{[k]}$ is either 0 or 1 depending on the actual outcome of this specific instance. Nevertheless if the k'th position is not uniquely defined and there are $m_{k,1}$ instances that can be located in this position, and $m_{k,2}$ of which are truly positive, then:

$$t^{[k]} = m_{k,2} \Big/ m_{k,1} \tag{5}$$

The sum of $t^{[k]}$ over the entire test set is equal to the number of instances that are labeled "positive". Moreover Hit-Rate(j) ≈ Precision ($p^{[j]}$) where $p^{[j]}$ denotes the jth order of $\hat{P}_I(\text{"}positive\text{"}|x_1),\ldots,\hat{P}_I(\text{"}positive\text{"}|x_m)$. The values are strictly equal when the value of ^{j}th is uniquely defined. It should be noticed that the hit-rate measure was originally defined without any reference to the uniqueness of certain position. However there are some classifiers that tend to provide the same conditional probability to several different instances. For instance in a decision tree, any instances in the test set that belongs to the same leaf get the same conditional probability. Thus, the proposed correction is required on those cases. Figure 2 illustrates a hit-curve.

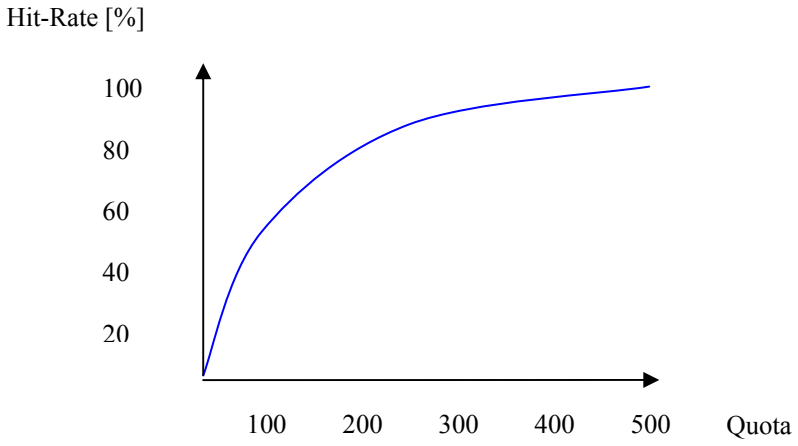

Figure 2: A typical hit curve.

2.2.4 Qrecall (Quota Recall)

The hit-rate measure, presented above, is the "precision" equivalent for quota-limited problems. Similarly, we suggest the Qrecall (for Quota Recall) to be the "recall" equivalent for quota-limited problems. The Qrecall for a certain position in a ranked list is calculated by dividing the number of positive instances, from the head of the list until that position, by the total positive instances in the entire dataset. Thus, the Qrecall for a quota of j is defined as:

$$\text{Qrecall}(j)=\frac{\sum_{k=1}^{j} t^{[k]}}{n^{+}} \qquad (6)$$

The denominator stands for the total number of instances that are classified as "positive" in the entire dataset, formally it can be calculated as:

$$n^{+} = \left|\{< \boldsymbol{x}_i, y_i > \; : \; y_i = "positive"\}\right| \qquad (7)$$

2.2.5 Lift Curve

A popular method of evaluating probabilistic models is *Lift* (Coppock, 2002). A ranked test set is divided into several portions (usually deciles). Lift is calculated as follows: the ratio of really positive instances in a specific decile divided by the average ratio of really positive instances in the population. Regardless of how the test set is divided, a good model is achieved if the lift decreases when proceeding to the bottom of the scoring list. A good model would present a lift greater than 1 in the top deciles and lift smaller than 1 in the last deciles. Figure 3 illustrates a lift chart for a typical model prediction. A comparison between models can be done by comparing the lift of the top portions, depending on the resources available and cost/benefit considerations.

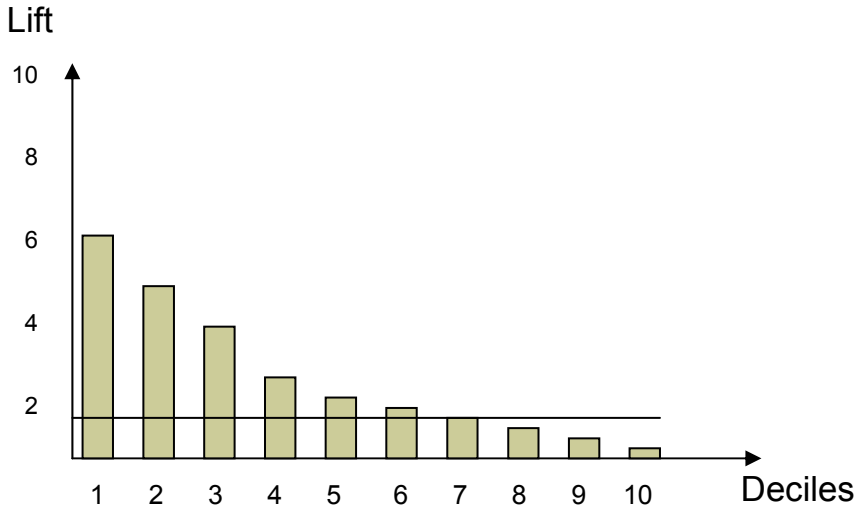

Figure 3 :A typical lift chart.

2.2.6 Pearson Correlation Coefficient

There are also some statistical measures that are candidates to be used as performance evaluators of models. These measures are known in the statistical literature for quite some time and can be found in many statistical books (see for instance Siegel, 1956). In this work we examine the *Pearson* correlation coefficient. This measure can be used to find the correlation between the ordered estimated conditional probability ($p^{[k]}$) and the ordered actual expected outcome ($t^{[k]}$). Pearson correlation coefficient can have any value between -1 and 1 where the value 1 represents the strongest positive correlation. It should be noticed that this measure take into account not only the ordinal place of an instance but also its value (i.e. the estimated probability attached to it). The *Pearson* correlation coefficient for two random variables is calculated by dividing the co-variance by the product of both standard deviations. In this case the standard deviations of the two variables assuming a quota size of j are:

$$\sigma_p(j)=\sqrt{\frac{1}{j}\sum_{i=1}^{j}\left(p^{[i]}-\bar{p}(j)\right)} \quad ; \quad \sigma_t(j)=\sqrt{\frac{1}{j}\sum_{i=1}^{j}\left(t^{[i]}-\bar{t}(j)\right)} \tag{8}$$

where $\bar{p}(j),\bar{t}(j)$ represent the average of $p^{[i]}$'s and $t^{[i]}$'s respectively:

$$\bar{p}(j)=\frac{\sum_{i=1}^{j}p^{[i]}}{j} \quad ; \quad \bar{t}(j)=\frac{\sum_{i=1}^{j}t^{[i]}}{j}=HitRate(j) \tag{9}$$

The co-variance is calculated as following:

$$Cov_{p,t}(j)=\frac{1}{j}\sum_{i-1}^{j}\left(p^{[i]}-\bar{p}(j)\right)\left(t^{[i]}-\bar{t}(j)\right) \tag{10}$$

Thus, the Pearson correlation coefficient for a quota j, is:

$$\rho_{p,t}(j)=\frac{Cov_{p,t}(j)}{\sigma_p(j)\cdot\sigma_t(j)} \tag{11}$$

2.3 Measures for evaluating limited and unknown in advance quota

The issue becomes more complicated when there is no specific quota to fill, but rather the interest is on the general performance of a classifier over an average quota or different sizes of quotas. Evaluating a probabilistic model without using a specific fixed quota is not a trivial task.

Using continuous measures like *hit curves, ROC curves* and *lift charts* that were mentioned previously, is problematic. Such measures can give a definite answer to the question: "which is the best model?" only if one model dominates in the curve space, meaning that all the other model's curves are beneath it or equal to it over the entire chart space. If a dominating model does not exist, than there is no answer to that question, using only this measure. Complete order demands no intersections of the curves. Of course, in practice there is almost never one dominating model. The best answer that can be obtained is in which areas one model outperforms the others. As shown in Figure 4: Every model gets different values in different areas. If a complete order of models performance is needed, another measure should be used.

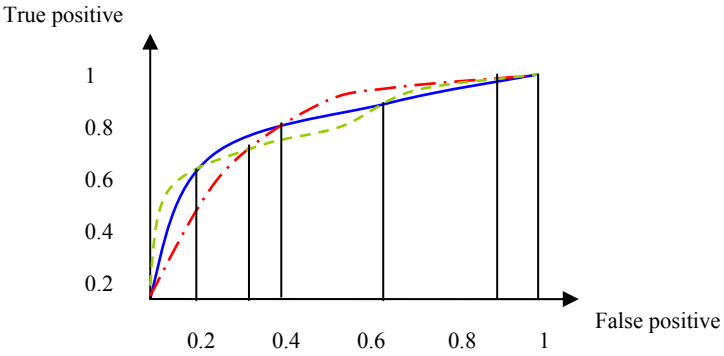

Figure 4 :Areas of dominancy. A ROC curve is an example to a measure that gives areas of dominancy and not a complete order of the models. In this example the equally dashed line model is the best for f.p (false positive) < 0.2. The full line model is the best for 0.2 < f.p <0.4. The dotted line model is best for 0.4 < f.p <0.9 and from 0.9 to 1 again the dashed line model is the best.

2.3.1 Area Under Curve (AUC)

Area Under the ROC Curve (AUC) is a useful metric for classifier performance as it is independent of the decision criterion selected and prior probabilities. The AUC comparison can establish a dominance relationship between classifiers. If the ROC curves are intersecting, the total AUC is an average comparison between models (Lee, 2000). The bigger it is, the better the model is. As opposed to other measures, the area under the ROC curve (AUC) does not depend on the imbalance of the training set (Kolcz *et al.*, 2003). Moreover Bradley (1997) argues the comparison of the AUC of to two classifiers is more fair and informative, than comparing their misclassification rates.

2.3.2 Average Hit Rate

Average hit rate is a weighted average of all hit-rate values. If the model is optimal, then all the really positive instances are located in the head of the ranked list, and the value of the average hit rate is 1. The use of this measure fits an organization that needs to minimize type II statistical error (namely including a certain object in the quota although in fact this object will be labeled as "negative"). Formally the Average Hit Rate for binary classification problems is defined as:

$$AverageHitRate = \frac{\sum_{j=1}^{n} t^{[j]} \cdot HitRate(j)}{n^{+}} \qquad (12)$$

where $t^{[j]}$ is defined as in Equation 4 and is used as weighting factor. Note that the average hit-rate ignores all hit-rate values on unique positions that are actually labeled as "negative" class (because $t^{[j]}$ =0 in these cases).

2.3.3 Average Qrecall

Average Qrecall is the average of all the Qrecalls which start from the position that is equal to the number of positive instances in the test set, to the bottom of the list. Average Qrecall fits an organization that needs to minimize type I statistical error (namely not including a certain object in the quota although in fact this object will be labeled as "positive"). Formally Average Qrecall is defined as:

$$\frac{\sum_{j=n^{+}}^{n} Qrecall(j)}{n - (n^{+} - 1)} \qquad (13)$$

where n is the total number of instances and n^{+} is defined in Equation (7).

Table 2 illustrates the calculation of Average Qrecall and Average Hit-rate for a dataset of ten instances. The table presents a list of instances in descending ordered descending according to their predicted conditional probability to be classified as "positive". Because all probabilities are unique, the third column ($t^{[k]}$) indicates the actual class ("1" represent "positive" and "0" represents "negative"). The average values are simple algebraic average of the highlighted cells.

Table 2: An example for calculating Average Qrecall and Average Hit-rate

Place in list (j)	*Positive probability*	$t^{[k]}$	*Qrecall*	*Hit rate*
1	0.45	1	0.25	1
2	0.34	0	0.25	0.5
3	0.32	1	0.5	0.667
4	0.26	1	0.75	0.75
5	0.15	0	0.75	0.6
6	0.14	0	0.75	0.5
7	0.09	1	1	0.571
8	0.07	0	1	0.5
9	0.06	0	1	0.444
10	0.03	0	1	0.4
		Average:	**0.893**	**0.747**

The different behavior of Qrecall and *Hit-rate* can be seen in Figure 5, which describes the values of the measures on Y axis versus the number of instances in a quota on the X axis. The values for the chart are taken from Table 2. Note that both *average Qrecall* and *average Hit-rate* gets the value 1 in an optimum classification, where all the positive instances

are located in the head of the list. This case is illustrated in Table 3. A summary of the key differences are provided in Table 4.

Table 3 :Qrecall and Hit-rate in an optimum prediction

Place in list (j)	*Positive probability*	$t^{[k]}$	*Qrecall*	*Hit rate*
1	0.45	1	0.25	1
2	0.34	1	0. 5	1
3	0.32	1	0.75	1
4	0.26	1	1	1
5	0.15	0	1	0.8
6	0.14	0	1	0.667
7	0.09	0	1	0.571
8	0.07	0	1	0.5
9	0.06	0	1	0.444
10	0.03	0	1	0.4
		Average:	**1**	**1**

Table 4: Characteristics of Qrecall and Hit-rate.

Parameter	Hit-rate	Qrecall
Function increasing/decreasing	Non monotonic	Monotonically increasing
End point	Proportion of positive samples in the set	1
Sensitivity of the measures value to positive instances	Very sensitive to positive instances at the top of the list. Less sensitive on going down to the bottom of the list.	Same sensitivity to positive instances in all places in the list.
Effect of negative class on the measure	A negative instance affects the measure and makes its value to decrease.	A negative instance does not affect the measure.
Range	0≤ Hit-rate ≤1	0≤ Qrecall ≤1

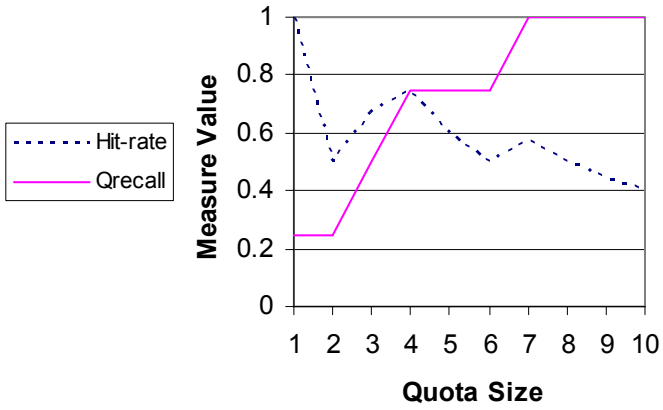

Figure 5 :Behavior of Qrecall and Hit-rate

2.3.4 PEM (Potential Extract Measure)

To better understand the behavior of Qrecall curves, consider the cases of random prediction and optimum prediction.

- Suppose no learning process was applied on the data and the list produced as a prediction would be the test set in its original (random) order. Under the assumption that positive instances are distributed uniformly in the population, then a quota of random size contains a number of positive instances that is proportional to the a-priory proportion of positive instances in the population. Thus, a Qrecall curve that describes a uniform distribution (which can be considered as a model that predicts as well as a random guess, without any learning) is a linear line (or semi linear because values are discrete) which starts at 0 (for zero quota size) and ends in 1.

- Suppose now that a model gave an optimum prediction, meaning all positive instances are located at the head of the list and below them all the negative instances. In that case the Qrecall curve climbs linearly until a value of 1 is achieved at point n^+ (n^+ = number of positive samples). From that point any quota that has a size bigger than n^+, fully extracts test set potential and the value 1 is kept until the end of the list.

Note that a "good model", which outperforms random classification, though not an optimum one, will be "on average" between these two curves. It may drop sometimes below the random curve but generally, more area is delineated between the "good model" curve and random curve, above the latter then below it. If the opposite is true then the model is a "bad model" that does worse than a random guess. Figures 6 and 7 give an intuition of the above. The examples relate to a test set of 100 instances, 20 of them positive and 80 negative.

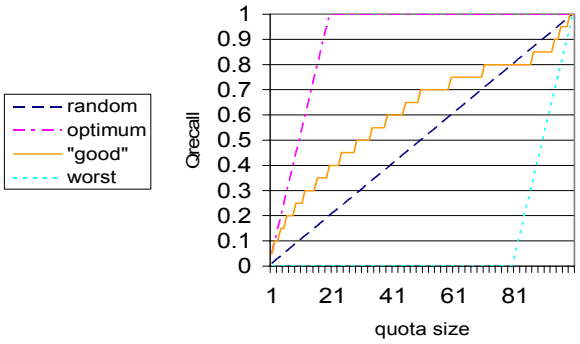

Figure 6: An example of a "good" model.

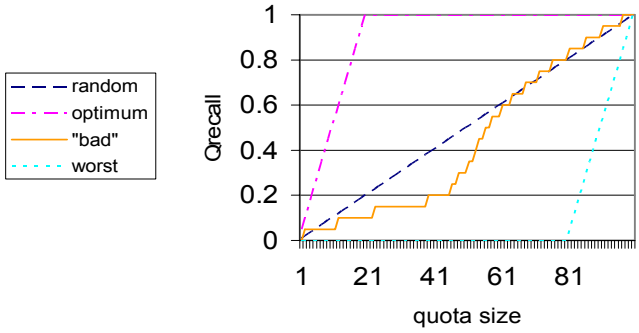

Figure 7 :An example of a "bad" model.

The last observation leads us to consider a measure that evaluates the performance of a model by summing the areas delineated between the Qrecall curve of the examined model and the Qrecall curve of a random model (which is linear). Areas above the linear curve are added and areas below the linear curve are subtracted. The areas themselves are calculated by subtracting the Qrecall of a random classification from the Qrecall of the model's classification in every point as shown in Figure 8. The areas where the model performed better than a random guess increase the measure's value while the areas where the model performed worse than a random guess decrease it. If the last total computed area is divided in the area delineated between the optimum model Qrecall curve and the random model (linear) Qrecall curve, then it reaches the extent to which the potential is extracted, independently of the number of instances in the dataset. Formally, the PEM (Potential Extract Measure) measure is calculated as:

$$PEM = \frac{S_1 - S_2}{S_3} \tag{14}$$

where S_1 is the area delimited by the Qrecall curve of the examined model above the Qrecall curve of a random model. S_2 is the area delimited by the Qrecall curve of the examined model under the Qrecall curve of a random model. S_3 is the area delimited by the optimal Qrecall curve and the curve of the random model. The division in S_3 is required in order to normalize the measure, thus datasets of different size can be compared. In this way, if the model is optimal, then PEM gets the value 1. If the model is as good as a random choice, then the PEM gets the value 0. If it gives the worst possible result (that is to say, it puts the positive samples in the bottom of the list), then its PEM is -1. Based on the notations defined above the PEM can be formulated as:

$$PEM = \frac{S_1 - S_2}{S_3} = \frac{\sum_{j=1}^{n} qrecall(j) - \sum_{j=1}^{n} \frac{j}{n}}{\sum_{j=1}^{n^+} \frac{j}{n^+} - \sum_{j=1}^{n} \frac{j}{n}} = \frac{\sum_{j=1}^{n} (qrecall(j)) - \frac{(n+1)}{2}}{\frac{(n^+ + 1)}{2} + n^- - \frac{(n+1)}{2}} = \frac{\sum_{j=1}^{n} (qrecall(j)) - \frac{(n+1)}{2}}{\frac{n^-}{2}} \quad (15)$$

where n^- denotes the number of instances that are actually classified as "negative". Table 5 illustrates the calculation of PEM for the instances in Table 2. Note that the random Qrecall does not represent a certain realization but the expected values. The optimal qrecall is calculated as if the "positive" instances have been located in the top of the list.

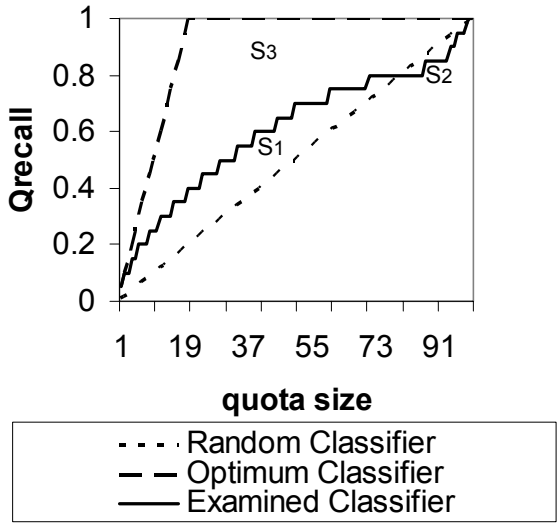

Figure 8: A qualitative representation of PEM.

Table 5: An example for calculating PEM for instances of Table 2.

Place in list	*Success probability*	$t^{[k]}$	*Model Qrecall*	*Random Qrecall*	*Optimal Qrecall*	*S1*	*S2*	*S3*
1	0.45	1	0.25	0.1	0.25	0.15	0	**0.15**
2	0.34	0	0.25	0.2	0.5	0.05	0	**0.3**
3	0.32	1	0.5	0.3	0.75	0.2	0	**0.45**
4	0.26	1	0.75	0.4	1	0.35	0	**0.6**
5	0.15	0	0.75	0.5	1	0.25	0	**0.5**
6	0.14	0	0.75	0.6	1	0.15	0	**0.4**
7	0.09	1	1	0.7	1	0.3	0	**0.3**
8	0.07	0	1	0.8	1	0.2	0	**0.2**
9	0.06	0	1	0.9	1	0.1	0	**0.1**
10	0.03	0	1	1	1	0	0	**0**
					Total	**1.75**	**0**	3

Note that the PEM somewhat resembles the Gini index produced from Lorentz curves which appear in economics when dealing with the distribution of income. Indeed this measure indicates the difference between the distribution of positive samples in a prediction and the uniform distribution. Note also that this measure gives an indication of the total lift of the model in every point. In every quota size, the difference between the Qrecall of the model and the Qrecall of a random model expresses the lift in extracting the potential of the test set due to the use in the model (for good or for bad).

2.3.5 Which measure should be used?

Average Hit-rate provides an answer to the question: "which classifier grants the best ratio of relevant instances over the total number of instances in an 'average quota'?". The answer to this question suits a decision maker that needs to fill a

flexible quota and the cost of accepting an irrelevant instance is high. It may be useful for decision maker that needs to minimize Type 2 statistical error.

An alternative question that might suits the needs of the decision maker is: "which classifier grants the best ratio of relevant instances over the total number of relevant instances available in the data set?". This question fits cases in which the cost of not accepting a relevant instance is high. It may be useful for decision maker that needs to minimize Type 1 statistical error. Such a decision maker would prefer to use average Qrecall or PEM.

This claim can be shown mathematically. As depicted in Figure 9, the line (c0) indicates the instances that are truly negative and the line (c1) indicates the instances that are truly positive. On the X axis is the estimated probability of an instance, according to a model's prediction. The threshold for a quota was set to c (cut off point). All the instances of c1 that got probability lower than c are out of the quota and therefore are false negative. All the instances of c0 that got probability greater than c are inside the quota and therefore are false positive.

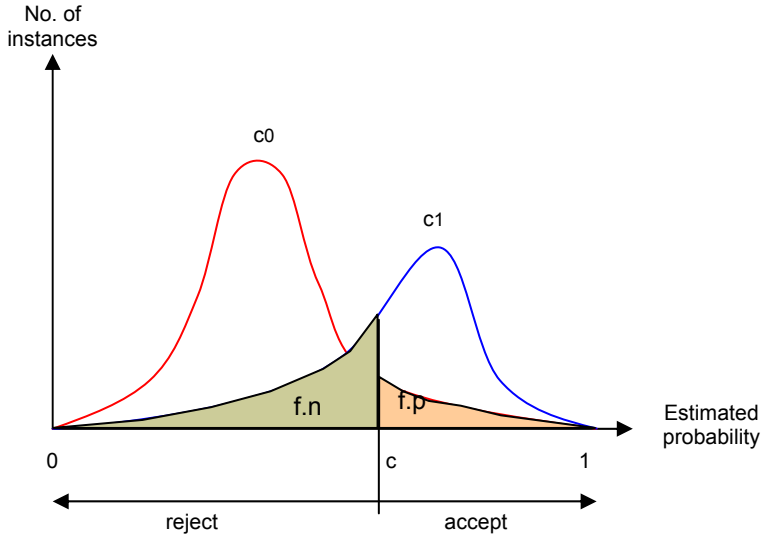

Figure 9 :False negative and false positive errors in a model's prediction ordered in a ranked list. The red line (c0) indicates the negative instances and the blue line (c1) indicate the positive instances.

Recall from Table 1 and note that:
α = Type 1 statistical error.
β = Type 2 statistical error.
c1 = describes the instances that really have positive class.
c0 = describe the instances that really have negative class.
r.p = the number of real positive instances (c+d in Table 1).
r.n = the number of real negative instances (a+b in Table 1).
f.n = the number of rejected instances under c1 (c in Table 1).
f.p = the number of accepted instances under c0 (b in Table 1).
p.p = the number of instances that were predicted positive (b+d in Table 1).

Note that the connections are:
α = f.n/r.p (16)
β = f.p/r.n (17)

Based on the precision definition in Equation 1:

$$1-precision=1-\frac{d}{b+d}=\frac{b+d-d}{b+d}=\frac{b}{b+d}=\frac{f.p}{p.p}=\frac{\beta*r.n}{p.p} \qquad (18)$$

This is the percentage of instances that were accepted and do not fit (relative area of f.p under all the accepted areas of c1 and c0 in Figure 9). Therefore high precision leads to a low β.

On the other hand:

$$1-recall=1-\frac{d}{c+d}=\frac{c+d-d}{c+d}=\frac{c}{c+d}=\frac{f.n}{r.p}=\alpha \qquad (19)$$

This is the percentage of instances that fit and were not accepted (relative area of f.n under c1 in Figure 9).
Thus, high recall leads to a low α

4. Experimental Study

4.1 Objectives

So far, several evaluation measures were presented. This section reviews experiments in which those measures were used. The experiments were designed to examine whether all the measures that were designed to evaluate a probabilistic classification according to its success, may provide a fair tool to use, no matter what the goals of the process are. Yet, is it important to define what type of error should be minimized in order to choose the most suitable measure for the task? Does it influence in any way on the achieved results? Moreover, the experiments should study the differences and similarities of the measures behaviors, in order to determine whether one of the measures can be used instead of another, even though they were developed in different ways and on different background. Particularly, it is interesting to examine the similarity of the novel measures suggested in this paper to the well validated, parametric measure: Pearson Correlation Coefficient and AUC measure.

The experimental study contains two parts. The first part introduce a real-world case study in which the proposed measures were used to evaluate the performance of models that were applied during an ensemble classification process. The second part validates the results on 20 datasets obtained from UCI repository.

4.2 Test Case

The test case that was used for the experiments was taken from the field of human resources. A company recruits each year several employees for a job. The job is very complex and requires considerable mechanical and cognitive skills. Hence, the training period for the job is very long and expensive. Since the qualifications needed for the job are compound, only a few applicants fit the company's needs. In order to save money and training time of applicants who will not complete the training period, the company prefers that only the applicants with the best chances for completing the training period will begin it.

One of the ways that the company uses to screen the applicants is by giving them missions in a simulator to check their skills in different scenarios. Each such mission creates a dataset in which each row represents an applicant and each column is a feature that represents the performance of the applicant in one of the skill parameters that is being checked. Some of the features contain continuous numbers and some are binary features. In addition, there are features that are transformations of other features in the dataset. The target attribute is a binary feature that has two classes: 1, if the applicant finished the training period and 0, if the applicant failed to complete his training period. Overall there are 52 datasets that contain past information about applicants that have already finished or failed to finish their training period. Each one represents a mission and contains the data of the applicants in a specific scenario. The scenarios are varied and check different skills.

Each one of the missions can be considered as a sensor that provides information about the applicant in certain areas which it was designed to test. Since the goal is to evaluate the overall performance of an applicant in all the areas in order to determine his chances of succeeding, this problem can be regarded as a sensor fusion problem.

It should be mentioned that the data was part of an experiment in which applicants that did not do well in the tests were allowed to continue the training period to see if they would succeed or fail. Since no one involved knew about their test performance scores, there is no possibility that self-fulfilling wishes substantially changed the results.

4.2.1 Experiment Design

The measures were tested through a data mining technique called selective voting. In selective voting the first best X models are taken and their results are combined by simple average. By increasing X from 1 to 52, a graph of the performance of the integrated prediction as a function of X is achieved. Figure 10 a-d illustrates the results of 4 experiments. In each experiment, models were created from a training set which included 250 instances and tested on a test set which included 50 instances. The ratio of positive instances out of the total instances in both sets was 0.2. Each one of the experiments included different instances in the test set. On the X axis are marks of 52 predictions made by 52 models. Every prediction is basically a scoring list of the instances of the test set, ordered by their estimated probability to finish the training period of the company. Each such prediction got 4 marks, which are presented on the Y axis, according to its success. The marks were given in average Hit-rate, average Qrecall, PEM and Pearson correlation coefficient. The scores provided by average Hit-rate and average Qrecall were normalized.

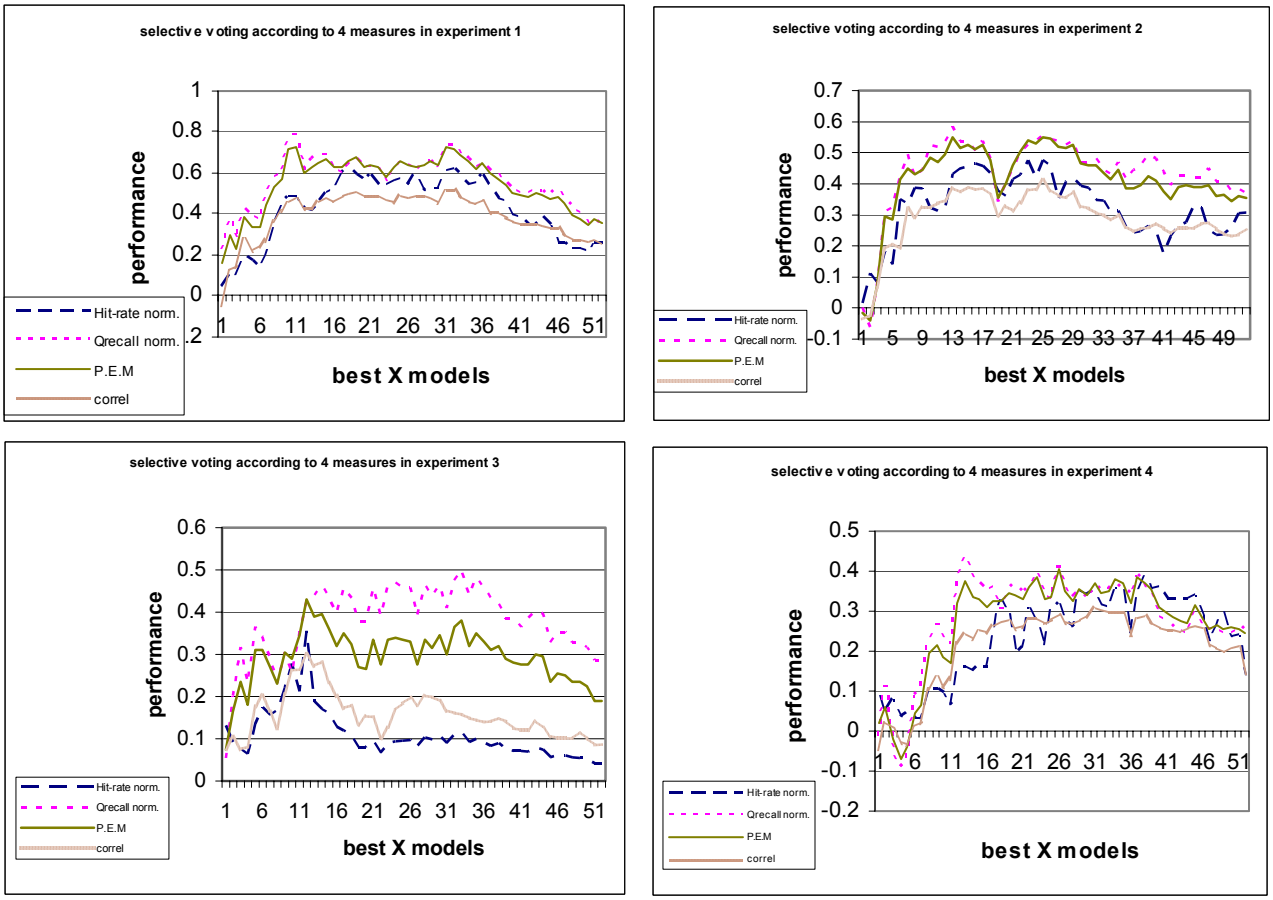

Figure 10a-d :Marks of 52 predictions of a scoring list, according to 4 measures

4.2.2 Comparison between performance measures

Looking at Figure 10 a-d, the next notions can be made:

- All measures have, most of the time, the same tendency. They increase and decrease simultaneously. This means that using any of those measures, with no respect to the type of error that is needed to be minimized will not cause a huge error. This outcome was expected for the first three evaluation measures because all of them were designed to render high marks if positive instances are at the top of a list and vice versa. It is encouraging to see, though, that the Pearson Correlation Coefficient displays a similar behavior.
- The measures reach a global maximum at different points on the graphs. As can be seen in Table 6. This means that despite the resembling tendency of the measures, the exact measure to use is significant for achieving optimality according to specific objectives.

Table 6: Points on the graphs, where global optimum was achieved.

	Experiment 1	Experiment 2	Experiment 3	Experiment 4
Average Hit-rate	32	23	12	38
Average Qrecall	11	13	33	13
PEM	11and 31	13 and 25	12	26
Pearson Correlation Coefficient	32	25	12	31

Remarks:

- The absolute numbers that a prediction gets is not important. When deciding between alternatives, what matters is the relative mark that a prediction got in comparison to other predictions using the same measure.
- The fact that for a certain prediction, a mark given by one measure is higher than a mark given by another measure, does not mean that the higher measure is better. The measures measure different things and therefore can not be compared by their absolute values.

Table 7 and Figure 11 show the correlation between the marks that were given by the different measures. The correlation was checked on the results presented in Figure 10 a-d. Highlighted are the highest results (correlation greater

than 0.95). It can be noticed that average Qrecall is highly correlated to PEM. There is no wonder about it because PEM is based on Qrecall curves. Another notion is that Pearson correlation coefficient has high correlation with PEM and they behave in a resembling manner. This notion has a very high value since it means two things:

1. PEM provides similar results to a traditional, high validated measure from the field of parametric statistics. In some aspects it makes this measure, which has not yet been tested and gained validity, more "trustable".
2. It also means that Pearson correlation coefficient, which was not designed for purposes of evaluating the performance of probabilistic classification of models, can be in fact used for that task exactly and provide very good results.

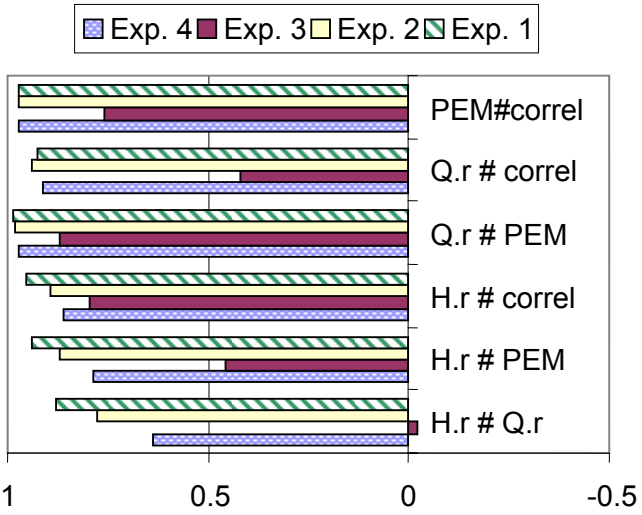

Figure 11: Correlation between measures in selective voting

Table 7: Correlation between measures in selective voting

measures	Exp. 1	Exp. 2	Exp. 3	Exp. 4
H.r # Q.r	0.878913	0.779044	-0.01978	0.639342
H.r # PEM	0.941348	0.871575	0.457021	0.78519
H.r # correl	0.951883	0.892064	0.79621	0.859306
Q.r # PEM	0.98482	0.982845	0.870869	0.974288
Q.r # correl	0.92394	0.939599	0.419468	0.912067
PEM#correl	0.970244	0.971571	0.758552	0.971114

4.3 UCI Repository Datasets

In order to examine the potential of the proposed measures on various classification problems, a comparative experiment has been conducted. The following subsections describe the experimental set-up and the obtained results.

4.3.1 Experiment Design

This experiment uses three different types of probabilistic classifiers, more specifically: C4.5, Naïve Bayes and Neural Networks. The C4.5 algorithm was chosen because it is considered as the state-of-the-art decision tree algorithm which is widely used in many other comparative studies. Naïve Bayes was chosen because it is considered as a simple but yet efficient classifier. Neural Networks was chosen due its popularity.

The selected algorithms were examined on 20 datasets which have been selected manually from the UCI Machine Learning Repository (Merz and Murphy, 1998). The datasets chosen vary across a number of dimensions such as: the number of classes, the number of instances, the number of input features and their type (nominal, numeric).

We randomly split the dataset into a training set (2/3 of the instances) and a test set (1/3 of the instances). We have trained the classifier using the training set and evaluate each measure on the test set. The experiment was conducted using the Weka framework (Witten and Frank, 2005). For this purpose we have extended its regular evaluation methods to measure the new proposed measures (average hit-rate, average qrecall and PEM).

4.3.2 Comparison between performance measures

Table 8 and Table 9 present the obtained results for the smallest class and the largest class respectively. In most of the cases all measures lead to the same conclusion regarding the performance of each classifier relative to the other classifiers. This result is consistent with the results that were obtained in the abovementioned test case.

When comparing the AUC measure and the PEM measure the following observations can be identified:

1. Both measures obtain the maximum value (1) together.
2. In binary problems, AUC always gets the same value for both classes. In PEM this is not necessarily true. For instance for the Labor dataset, the values are equal when Naïve Bayes is used, but these values are not equal when C4.5 is used. In fact for Naïve Bayes in all binary cases the values of PEM measures are identical. In Neural Networks 11 of 12 cases this observation is true. However for C4.5 this observation was true only for the mushroom dataset. Note that the mushroom dataset is the biggest binary training set. This behavior of C4.5 can be explained by the fact that in this decision trees, it is more frequent to have two different instances with the same predicted probability (as both instances are affiliated to the same leaf)
3. It is interesting to notice that although the PEM and AUC measures get different values, still using them as a method for selecting the best classifier among C4.5, Naïve Bayes and Neural Networks, result always with the same decision. Even when there quite small differences between the classifiers (see for instance the difference between NN and DT on chess dataset), still the decisions based on AUC and PEM were consistent. In the smallest class of the breast cancer dataset both Naïve Bayes and Neural Networks got the same AUC value but the PEM value of Naïve Bayes was a bit higher. A closer look has indicated that in this case the accuracy of Naïve Bayes was 71.4286 % while the accuracy of Neural Networks was only 70.4082 %. Thus, PEM is potentially more sensitive. However it is important to note that using average hit rate in the smallest class (Table 8) would leads to a different conclusion in this case.

When comparing the Average QRecall with PEM, it seems that not in all cases do these measures leads to the same decision regarding which the best classifier is (for instance Aust Credit, Audiology or Monks3 regarding the largest class). Thus, although PEM measure is based on Qrecall curves, it still provides new perspective for the decision maker.

Table 8: Summary of experimental results for the smallest class.

Dataset	# Instances	# Attributes	# Classes	C4.5				NB				NN		
				AUC	Average HitRate	Average Qrecall	PEM	AUC	Average HitRate	Average Qrecall	PEM	AUC	Average HitRate	Average Qrecall
Aust	690	15	2	0.864	0.903	0.912	0.747	0.9317	0.919	0.977	0.8634	0.9014	0.918	0.941
Audiology	226	70	24	1	1	1	1	1	1	1	1	1	1	1
Breast .Ca	286	10	2	0.604	0.468	0.703	0.158	0.676	0.561	0.779	0.351	0.676	0.616	0.751
Hepatitis	155	20	2	0.7184	0.319	0.7803	0.4040	0.8914	0.7116	0.9267	0.7828	0.9217	0.7245	0.952
Iris	150	5	3	0.967	0.955	0.968	0.927	0.987	0.980	0.993	0.975	0.998	0.996	1.0
Kr-vs-kp	3196	37	2	0.9981	0.9983	0.9984	0.9958	0.9217	0.9189	0.9649	0.8433	0.9992	0.9993	0.9993
Labor	57	17	2	0.835	0.892	0.890	0.736	0.978	0.968	0.989	0.956	1	1	1
Lung Ca.	32	57	3	0.7321	0.4583	0.8214	0.1428	0.75	0.7611	0.8571	0.5	0.7143	0.5666	0.8928
Monks1	124	7	2	0.7413	0.7707	0.8701	0.5064	0.8853	0.91	0.93	0.7705	1	1	1
Monks2	169	6	2	0.5251	0.3863	0.74	0.0579	0.4878	0.358	0.688	0.0244	1	1	1
Monks3	122	6	2	0.9841	0.9923	0.9931	0.9818	0.9388	0.9604	0.9546	0.877	0.9365	0.958	0.9546
MUSH	8124	22	2	1	1	1	1	0.9978	0.997	0.999	0.9956	1	1	1
Nurse	12960	8	5	0.9533	0.7956	0.9544	0.9059	0.9956	0.8266	0.9975	0.9911	0.9996	0.9898	0.9996
OPTIC	5628	64	10	0.7027	0.46	0.8	0.5722	0.9853	0.8998	0.99	0.97	0.96	0.87	0.975
Pima I. Dia.	768	9	2	0.8137	0.647	0.875	0.616	0.8551	0.762	0.91	0.7101	0.7855	0.6741	0.855
Sonar	208	60	2	0.6774	0.7362	0.82	0.365	0.769	0.799	0.892	0.538	0.9563	0.968	0.968
Soybean	683	35	19	1	1	1	1	1	1	1	1	1	1	1
Vote	290	16	2	0.9899	0.99	0.987	0.972	0.9801	0.974	0.99	0.96	0.9991	0.9987	0.9998
Wine	178	13	3	0.963	0.902	0.991	0.932	1	1	1	1	1	1	1
Zoo	101	8	7	1	1	1	1	1	1	1	1	1	1	1

Table 9: Summary of experimental results for the largest class.

Dataset	# Instances	# Attributes	# Classes	C4.5				NB				NN			
				AUC	Average HitRate	Average Qrecall	PEM	AUC	Average HitRate	Average Qrecall	PEM	AUC	Average HitRate	Average Qrecall	PEM
Aust Cred	690	15	2	0.864	0.787	0.950	0.709	0.9317	0.938	0.955	0.8634	0.9014	0.85	0.961	0.8027
Audiology	226	70	24	0.9315	0.8956	0.937	0.8469	0.9923	0.9807	0.997	0.984	0.9881	0.9671	0.9982	0.9761
Breast .Ca	286	10	2	0.604	0.736	0.88	0.257	0.676	0.791	0.903	0.351	0.676	0.7514	0.919	0.326
Hepatitis	155	20	2	0.7184	0.936	0.9419	0.4696	0.8914	0.9762	0.9747	0.7828	0.9217	0.9841	0.9772	0.8434
Iris	150	5	3	1	1	1	1	1	1	1	1	1	1	1	1
Kr-vs-kp	3196	37	2	0.9981	0.9981	0.9994	0.9966	0.9217	0.931	0.957	0.8433	0.9992	0.9992	0.9999	0.9984
Labor	57	17	2	0.835	0.843	0.934	0.604	0.978	0.988	1	0.956	1	1	1	1
Lung Ca.	32	57	3	0.9375	0.9166	1	0.9166	0.9167	0.8055	1	0.8333	0.875	0.7555	0.9583	0.75
Monks1	124	6	2	0.7413	0.6976	0.8722	0.4588	0.8853	0.8557	0.9653	0.7705	1	1	1	1
Monks2	169	6	2	0.5251	0.6540	0.8146	0.0424	0.4878	0.673	0.815	0.0244	1	1	1	1
Monks3	122	6	2	0.9841	0.972	1	0.9546	0.9388	0.8957	0.99	0.877	0.9365	0.8811	0.9863	0.873
MUSH	8124	22	2	1	1	1	1	0.9978	0.9979	0.9987	0.9956	1	1	1	1
Nurse	12960	8	5	1	1	1	1	1	1	1	1	1	1	1	1
OPTIC	5628	64	10	0.8084	0.299	0.7365	0.41	0.9939	0.9485	0.9966	0.9877	0.963	0.924	0.9656	0.9621
Pima I. Dia.	768	9	2	0.8137	0.8978	0.935	0.638	0.8551	0.9191	0.9465	0.7101	0.7855	0.869	0.932	0.571
Sonar	208	60	2	0.6774	0.6411	0.8436	0.3444	0.769	0.7582	0.886	0.538	0.9563	0.9435	0.99	0.912
Soybean	683	35	19	0.989	0.945	0.996	0.982	0.984	0.924	0.989	0.968	0.981	0.942	0.984	0.962
Vote	290	16	2	0.9899	0.995	0.998	0.987	0.9801	0.986	0.991	0.96	0.9991	0.999	0.9996	0.9981
Wine	178	13	3	0.859	0.759	0.918	0.708	0.9942	0.991	0.995	0.988	0.9965	0.9941	0.9988	0.993
Zoo	101	8	7	1	1	1	1	1	1	1	1	1	1	1	1

5. Conclusions

Real life problems are usually subject to resources constraints. In these cases, in order to obtain an effective classification process that can lead to effective decisions, a probabilistic classification must be used. This kind of classification obligates the use of appropriate measures that can give an evaluation of the performance of a model. Without such measures no definite comparison can be made between models.

This paper suggests two new evaluation measures: Qrecall and PEM. The measures suggested here usually behave in a similar manner, but since they are designed for different tasks they reach optimality in different points and may differ in their "opinion" about the best prediction out of a given set of predictions. The experiments indicated that the measures do not act the same in all cases and there is a great importance to define first the targets for which the classification process is implemented and only then to decide which measure to use. The paper shows the relation between the types of statistic errors that should be minimized and the evaluation measure that should be used. More specifically, this paper concludes that when type-1 error is of interest, then Qrecall or PEM should be used, and when type-2 error is of concern, then hit rates should be used.

Another way to look at it is through the concepts of negative screening and positive screening. Negative screening means screening out all instances that have a low chance to belong to the class of interest. Positive screening means accepting all the instances that have sufficient chances to belong to the class of interest. Different situations require different kinds of screening. For example, in a progressive screening, where after each phase a smaller number of instances proceed to the next phase, one may encounter in the preliminary phases a negative screening, which purpose is to screen out all instances with zero probability of belonging to the class of interest. This kind of screening at this phase cut down expenses in the next phases without loosing potential instances. However, in the mature phases of the screening, a positive screening is more appropriate when a certain quota has to be filled. Generally, every measure can be used for both tasks. Yet, for negative screening that aims to keep potential instances while denying very low potential instances, measures like Qrecall and PEM might be more appropriate. For positive screening that is oriented to minimize the misses in a quota, Hit-rate might be more appropriate.

This paper also shows the usefulness of Pearson Correlation Coefficient, a traditional measure from the field of statistics for the purpose of evaluating a probabilistic classification and showing the similarity of its results to the results of the novel measures suggested here.

References

1. Ali K. M. and Pazzani M. J. (1996) “Error reduction through learning multiple descriptions”. *Machine Learning 24, pp. 173-202.*
2. A.P. Bradley (1997), The use of the area under the roc curve in the evaluation of machine learning algorithms. Pattern Recognition, 30(7):1145–1159.
3. An A. and Wang Y. (2001)”Comparisons of classification methods for screening potential compounds”. *In IEEE International Conference on Data Mining, 2001.*
4. Coppock D. S. (2002) "Data Modeling and Mining: Why Lift?" *Published in DM Review online, June 2002. http://www.dmreview.com/master.cfm?NavID=55&EdID=5329*
5. Elkan C. (2001)”Magical thinking in data mining: Lessons from Coil challenge 2000”. In *Proceedings of the Seventh International Conference on Knowledge Discovery and Data Mining (KDD'01), pp. 426-431.*
6. A. Kolcz, A. Chowdhury, and J. Alspector (2003). Data duplication: An imbalance problem "In Workshop on Learning from Imbalanced Data Sets" (ICML).
7. Kubat M. and Matwin S. (1997)”Addressing the curse of imbalanced training sets: one sided selection”. *Proceedings of the Fourteenth International Conference on Machine Learning. Morgan Kaufman, pp. 179-186.*
8. Lee, S.~S. (2000). Noisy {R}eplication in {S}kewed {B}inary {C}lassification, Computational Statistics and Data Analysis, 34.
9. Levin N., Zahavi J. (2005), Data Mining for Target Marketing, In O. Maimon and L. Rokach (eds.), Data Mining and Knowledge Discovery Handbook, Springer, pp. 1261-1300.
10. Provost F. and Fewcett T. and Kohavi R. (1998)”The case against accuracy estimation for comparing induction algorithms”. *Proceedings of the Fifteenth International Conference on Machine Learning*, *pp. 445-453.*
11. Provost, F. and Fawcett, T. (2001), Robust {C}lassification for {I}mprecise {E}nvironments, Machine Learning, 42/3:203-231.
12. Siegel S. (1956) Non parametric statistics for the behavioral sciences. *McGraw Hill Book Company, Inc.1956, pp.195-238.*
13. Weiss G. M. and Hirsh H. (2000) “Learning to predict extremely rare events.” *Papers from the AAAI Workshop on Learning from Imbalanced Data Sets*, Technical Report WS-00-05, AAAI Press, Menlo Park, CA, 64-68.
14. Weiss G. M. and Provost F. (2001)”The effect of class distribution on classifier learning”. *Technical Report ML-TR-43, Department of Computer Science, Rutgers University.*
15. Ian H. Witten and Eibe Frank (2005) "Data Mining: Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

Correction Report

# Reviewer	# Comment	Comment Description	Correction Description
1	1	The motivation and necessities are still unclear even though the authors explained it in the Introduction. MORE SPECIFIC examples should be provided for the two reasons. The readers who aren't familiar with this topic may not capture the key points.	The introduction has been revised and it includes 3 examples. Moreover the motivation has been revised to make it clearer regarding the contribution of this paper.
1	2	The paper is TOO verbose and flat. The orgarnization is not good. The paper should be revised to CLARIFY their points in many phrases and sentences.	A major revision in the structure and phrasing has been made. The measures are now organized into 3 categories (see page 2). Some measures (such as spearman rank) that are not relevant to the understanding of this paper have been removed. New notations are used in order to make all equations clearer and accurate.
1	3	The experiment is not sufficient as its current form. More experiments are required to justify the authors' suggestions.	We extended the experimental study (see section 4.3). This experiment uses three different types of inducers (C4.5, Naïve Bayes and Neural Networks). It include 20 datasets from the UCI repository. The datasets chosen vary across a number of dimensions such as: the number of classes, the number of instances, the number of input features and their type (nominal, numeric). We are also compare the results to the well-known AUC measure.
2	1	The quality of the report need to be improved by evaluating the performance measures for a classifier for different types of data sets (numerical, categorical, sequence etc) with analysis of the results. Even large training and test sets with more number of classes are to be considered.	We extended the experimental study (see section 4.3). This experiment uses three different types of inducers (C4.5, Naïve Bayes and Neural Networks). It include 20 datasets from the UCI repository. The datasets chosen vary across a number of dimensions such as: the number of classes, the number of instances, the number of input features and their type (nominal, numeric). We are also compare the results to the well-known AUC measure.
2	2	Check the formulas and limits of the variables provided	Done
3	1	In page 2, under table 2, following definitions should be given clearly, because they are used in later sections. True positive rate (recall) is: d/(c+d) False positive rate is: b/(a+b) True negative rate is: a/(a+b) False negative rate is: c/c+d	Done

3	2	In page 11, in the definition of Average Hit-rate, "j=1/tk=1" should be "j=1/tj=1".	Equation 12 on page 6. We are now using new notations in order to make all equations clearer and accurate.
3	3	In page 15, in the definition of PEM using Hit-rate and Qrecall, (n+1)/2 should be subtracted from denominator.	Done
3	4	In page 15, following definition should be given. Cw=the Qrecall curve of a worst prediction	We have removed the notation Ci in and explain the areas verbally on page 8.
3	5	In page 6, in "Pearson Correlation Coefficient", definition of mean and variance should be given	The explanation for the use of Pearson in this paper has been added. See equations 8 – 11.
3	6	In page 11, "i=model index" should be omitted	Done
3	7	The number of all formulas should be rearranged.	Done
3	8	In page 5, "2.2 Statistical Evaluation aMeasures" should be "2.2 Statistical Evaluation Measures".	In order to save place and make the paper more readable we have reorganized the measures. We have left only one statistical measure (Pearson), thus this correction is not relevant any more.
3	9	In page 15, "Co=the Qrecall curve of an optimal perdiction" should be "Co=the Qrecall curve of an optimal prediction".	We have removed the notation Ci in and explain the areas verbally on page 8.