Reliable Classifications with Machine Learning

Matjaž Kukar and Igor Kononenko

University of Ljubljana, Faculty of Computer and Information Science
Tržaška 25, SI-1001 Ljubljana, Slovenia
{matjaz.kukar,igor.kononenko}@fri.uni-lj.si

Abstract. In the past decades Machine Learning algorithms have been successfully used in numerous classification problems. While they usually significantly outperform domain experts (in terms of classification accuracy or otherwise), they are mostly not being used in practice. A plausible reason for this is that it is difficult to obtain an unbiased estimation of a *single classification's reliability*. In the paper we propose a general transductive method for estimation of classification's reliability on single examples that is independent of the applied Machine Learning algorithm. We compare our method with existing approaches and discuss its advantages. We perform extensive testing on 14 domains and 6 Machine Learning algorithms and show that our approach can frequently yield more than 100% improvement in reliability estimation performance.

1 Introduction

Usually Machine Learning algorithms output only bare classifications for the new unclassified examples. While there are ways for almost all Machine Learning algorithms to at least partially provide quantitative assessment of a classification in questions, so far there is no general method to assign reliability to a single classification. Note that we are interested in the classifier's performance on a single example and not in average performance on an independent dataset.

Let us define the reliability of classification as an estimated probability that the (single) classification is in fact the correct one. Some authors [16, 21] use for this purpose a statistical term *confidence*. We, however, have decided to use a term *reliability*, since its calculation and interpretation are not always strictly statistical. For a given example description x_i we define the reliability of its predicted class y_i as follows.

$$\mathrm{Rel}(y_i) = P(y_i \text{ is a true class of example } x_i) \tag{1}$$

There have been numerous attempts to assign probabilities to Machine Learning classifiers' (decision trees and rules, Bayesian classifiers, neural networks, nearest neighbour classifiers, ...) in order to interpret their decision as a probability distribution over all possible classes. In fact, we can trivially convert every Machine Learning classifier's output to a probability distribution by assigning the predicted class the probability 1, and 0 to all other possible classes. The posterior probability of the predicted class can be viewed as a classifier's trust in its prediction (reliability) [3, 19]. However, such estimations may not be good due to the applied algorithm's language and representational biases.

T. Elomaa et al. (Eds.): ECML, LNAI 2430, pp. 219–231, 2002.

There is some ongoing work for constructing classifiers that divide the data space into regions that are reliable and regions that are not reliable [1]. Such meta-learning approaches have also been used for picking the most reliable prediction from the outputs of an ensemble of classifiers [14, 17].

We propose a different approach based on a general transductive method for reliability estimations. Our approach differs from the above in the following:

- it does not divide the data space into reliable and unreliable regions, but works instead on single data points (examples),
- it does not induce a meta-classifier at all, but instead uses a transductive framework to generate a reliability estimate for each single example.

Our approach is independent of the applied Machine Learning algorithm and requires only that it is able to represent its classifications as probability distributions. The core idea is to compare differences in classification's probability distributions between inductive and transductive steps and use them to assess reliability of single points (examples) in data space. Such assessments are very useful, especially in risk-sensitive applications (medical diagnosis, financial and critical control applications) because there it often matters, how much one can rely upon a given prediction. In such cases a general reliability measure of a classifier (e.g. classification accuracy, mean squared error, ...) with respect to the whole input distribution would not provide desired warranty. Another use of reliability estimations is in ensembles for selecting or combining answers from different classifiers [8].

The paper is organized as follows. In Sec. 2 we describe the basic ideas of transductive inference and outline the reasons why transductive reliability estimation should work well. In Sec. 3 we develop our idea for general and efficient implementation of transductive reliability estimation. In Sec. 4 we evaluate our approach on 14 domains with 6 Machine Learning algorithms. In Sec. 5 we present some conclusions and directions for future work.

2 Transduction Principle for Reliability Estimation

Transduction is an inference principle that takes a training sample and aims at estimating the values of a discrete or continuous function only at given unlabelled points of interest from input space, as opposed to the whole input space for induction. In the learning process the unlabelled points are suitably labelled and included into the training sample. The usefulness of unlabelled data [12] has among others been advocated in the context of co-training. It has been shown that for a better-than-random [2] classifier its performance can be significantly boosted by using only additional unlabelled data.

It has been suggested [20] that when solving a given problem one should avoid solving a more general problem as an intermediate step. The reasoning behind this principle is that, in order to solve a more general task, resources may be wasted or compromises made which would not have been necessary for solving only the problem at hand (i.e. function estimation only on given points). This common-sense principle reduces a more general problem of inferring a functional dependency on the whole input space (inductive inference) to the problem of estimating the values of a function only at given points (transductive inference).

Let $\mathcal{X}$ be a space of attribute descriptions of points in a training sample, and $\mathcal{Y}$ a space of labels (continuous or discrete) assigned to each point. Given a probability distribution $\mathcal{P}$, defined on the input space $\mathcal{X} \times \mathcal{Y}$, a training sample

$$S = \{(x_1, y_1), \ldots, (x_l, y_l)\} \tag{2}$$

consisting of l points, is drawn i.i.d. (identically independently distributed) according to $\mathcal{P}$. Additional m data points (working sample)

$$W = \{x_{l+1}, \ldots, x_{l+m},\} \tag{3}$$

with unknown labels are drawn in the same manner. The goal of transductive inference is to label all the points from the sample W using a fixed set $\mathcal{H}$ of functions $f : \mathcal{X} \mapsto \mathcal{Y}$ in order to minimize an error functional both in the training sample S and in the working sample W (effectively, in $S \cup W$) [5, 16]. In contrast, inductive inference (excluding ensembles of classifiers) aims at choosing a single function $f \in \mathcal{H}$ that is best suited to the unknown probability distribution $\mathcal{P}$.

At this point arises a question how to calculate the labels for a working sample. This can be done by labelling every point from a working sample with every possible label value; however given m working points and n possible class labels this leads to a combinatorial explosion yielding n^m possible labellings. For each possible labelling, an induction process on $S \cup W$ is run, and an error functional (error rate) is calculated.

By leveraging the i.i.d. sampling assumption and transductive inference, one can for each labelling estimate its reliability (a probability that it is correct). If the i.i.d. assumption holds, the training sample S as well as the joint correctly labelled sample $S \cup W$ should both reflect the same underlying probability distribution $\mathcal{P}$.

If one could measure a degree of similarity between probability distributions $\mathcal{P}(S)$ and $\mathcal{P}(S \cup W)$, this could be used as a measure of reliability of the particular labelling. Unfortunately, this problem is in non-computable [11], so approximation methods have to be used [21, 9].

2.1 Why does Transduction Work?

There is a strong connection between the transduction principle and the algorithmic (Kolmogorov) complexity. Let the sets S and $S \cup W$ be represented as binary strings u and v, respectively. Let $l(v)$ be the length of the string v and $C(v)$ its Kolmogorov complexity. We define the *randomness deficiency* of the string v as following [11, 21]:

$$\delta(v) = l(v) - C(v) \tag{4}$$

Randomness deficiency measures how random is the respective binary string and therefore the set it represents. The larger it is, the more regular the string (and the set). If we could calculate the randomness deficiency (but we cannot, since it is not computable), we could do it for all possible labellings of the set $S \cup W$ and select the labelling of W that results in the largest randomness deficiency of the joint set $S \cup W$ as the most probable one [21]. We could also construct a universal Martin-Löf's test for randomness [11]:

$$\sum \{P(x|l(x) = n) : \delta(x) \geq m\} \leq 2^{-m} \tag{5}$$

That is, for all binary strings of fixed length n, the probability of their randomness deficiency δ being greater than m is less than 2^{-m}. The value $2^{-\delta(x)}$ is therefore a p-value function for our randomness test [21].

Unfortunately, the definition of randomness deficiency is based on the Kolmogorov complexity and is not computable. Therefore we need feasible approximations to use this principle in practice. Extensive work has been done by using Support Vector Machines [5, 16, 21], however no general approach exists so far.

2.2 A Machine Learning Interpretation

In Machine Learning terms, the sets S and $S \cup W$ are represented with induced models M_S and $M_{S \cup W}$. Randomness of the sets is reflected in the (Kolmogorov) complexity of the respective models. If for the set $S \cup W$ the labelling with the largest randomness deficiency is selected, it follows from the definition (Eq. 4) that since the uncompressed description length $l(v)$ is constant, the Kolmogorov complexity $C(M_{S \cup W})$ is minimal. This implies that the respective labelling of W is most consistent with the training data S, since the minimal Kolmogorov complexity implies most regularities in the data. This in order implies that our Machine Learning algorithm will induce the model $M_{S \cup W}$ that will be most similar to the M_S.[1] Ideally, if the training data S is sufficient for inducing a perfect model, there is no difference between M_S and $M_{S \cup W}$.

This greatly simplifies our view on the problem, namely it suffices to compare the (finite) models M_S and $M_{S \cup W}$. Greater difference means that the set $S \cup W$ is more random than the set S and (under the assumption that S is sufficient for learning effective model) that W consist of (at least some) improperly labelled, untypical examples.

Although the problem seems easier now, it is still a computational burden to calculate changes between model descriptions (assuming that they can be efficiently coded; black-box methods are thus out of question). However, there exists another way.

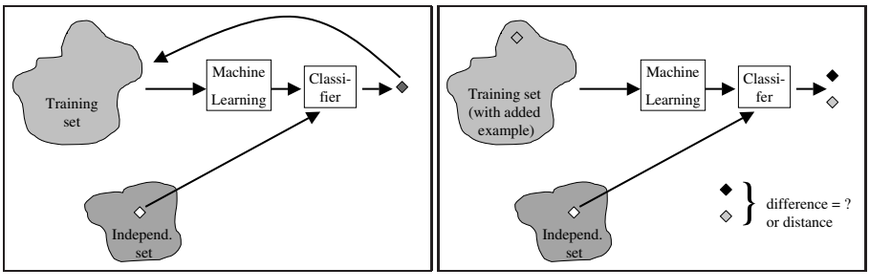
Since transduction is an inference principle that aims at estimating the values of a function only at given points of interest from input space (the set W), we are interested only in model change considering these examples. Therefore we can compare the classifications (or even better, probability distributions) of models M_S and models $M_{S \cup W}$. Obviously, the labelling of W that would minimally change the model M_S is as given by M_S. We will examine this approach in more detail in the next section.

3 Efficient Transductive Reliability Estimations

The prerequisite for a Machine Learning algorithm to be used in a transductive reliability framework is to represent its classifications as a probability distribution over all possible classes, although these distributions may not be very good estimates.

The transductive reliability estimation process is basically a two-step process, featuring an *inductive step* followed by a *transductive step*.

[1] Actually, here it would be more appropriate to use a prefix Kolmogorov complexity $K(\_)$ instead of $C(\_)$, and two-part MDL-style (model+exceptions) descriptions of the sets, since the Kolmogorov complexity $C(\_)$ itself is non-monotonic [11] wrt. the string length.



(a) Inductive step. (b) Transductive step.

Fig. 1. Transductive reliability estimation

- An *inductive step* is just like an ordinary inductive learning process in Machine Learning. A Machine Learning algorithm is run on the training set, *inducing* a classifier. A selected example is taken from an independent dataset and classified using the induced classifier. The same example is duplicated, labelled with its assigned class, and finally included into the training set (Fig. 1a).
- A *transductive step* is almost a repetition of an inductive step. A Machine Learning algorithm is run on the changed training set, *transducing* a classifier. The same example as before is taken from the independent dataset and again classified, now using the transduced classifier (Fig. 1b). Both classifications (represented by probability distributions) of the same example are compared and their difference (distance) is calculated, thus approximating the randomness deficiency. A brief algorithmic sketch is given in Fig. 2.

3.1 Calculating the Difference between Probability Distributions

Since a prerequisite for a Machine Learning algorithm is to represent its classifications as a probability distribution over all possible classes, we need a method to measure the difference between two probability distributions. The difference between two probability distributions (over discrete item sets of size $N < \infty$) can be viewed as a distance between two vectors in R^N. In principle, any *metric* can be used, however not all strict metric properties are required. We require only that the difference measure D between probability distributions P and Q satisfies the following:

1. $D(P,Q) \geq 0$ (nonnegativity)
2. $0 \leq D(P,Q) \leq \infty$, where $D(P,Q) = 0 \Leftrightarrow P = Q$
3. $D(P,Q) = D(Q,P)$ (symmetry law).

In our case P is a probability distribution after the inductive step, and Q is a probability distribution after the transductive step. For calculating the difference between probability distributions, a *Kullback-Leibler divergence* is frequently used [18]. In our experiments we use a symmetric Kullback-Leibler divergence.

Requires:	Machine Learning classifier, a training set and an unlabelled test example
Ensures:	Estimation of test example's classification reliability

1: Inductive step:
- train a classifier from the provided training set
- select an unlabelled test example and classify this example with an induced classifier
- label this example with a predicted class
- temporarily add the newly labelled example to the training set

2: Transductive step:
- train a classifier from the extended training set
- select the same unlabelled test example as above and classify this example with a transduced classifier

3: Calculate a randomness deficiency approximation as a *difference* between inductive and transductive classification.
4: Calculate the reliability of classification as $2^{-difference}$.

Fig. 2. The algorithm for transductive reliability estimation

3.2 Kullback-Leibler Divergence

Kullback-Leibler divergence, also frequently referred to as a relative entropy or I-divergence, is defined between probability distributions P and Q

$$I(P,Q) = -\sum_{i=1}^{n} p_i \log_2 \frac{p_i}{q_i} \tag{6}$$

Symmetric Kullback-Leibler divergence, or J-divergence, is defined between probability distributions P and Q

$$J(P,Q) \;=\; (I(P,Q)+I(Q,P)) \;=\; \sum_{i=1}^{n}(p_i-q_i)\log_2 \frac{p_i}{q_i} \tag{7}$$

$J(P,Q)$ is limited to the interval $[0,\infty]$, with $J(P,P)=0$. Similarly to the p-values of the universal Martin-Löf randomness test (Eq. 5), we calculate our reliability estimation as

$$\mathrm{Rel}(P,Q) = 2^{-J(P,Q)} \tag{8}$$

However, measuring the difference between probability distributions does not always perform well. There are at least a few exceptional classifiers (albeit trivial ones) where our original approach utterly fails.

3.3 The Curse of Trivial Models

So far we have implicitly assumed that the model used by the classifier is good (at the very least better than random). Unsurprisingly, our approach works very well with

random classifiers (probability distributions are randomly calculated) by effectively labelling their classifications as unreliable [8].

On the other hand, there also exist simple *constant* and *majority* classifiers. A *constant classifier* is such that it classifies all examples into the same class C_k with probability 1. In such cases our approach always yields reliability 1 since there is no change in probability distribution. A *majority classifier* is such that it classifies all examples into the same class C_k that is the majority class in the training set. Probability distribution is always the same and corresponds to the distribution of classes in the training set. In such cases our approach yields reliability very close to 1 since there is almost no change in probability distribution (only for the example in question), that is at most for $1/N$, where N is number of training examples. In large datasets this change is negligible.

Note that such extreme cases do occur in practice and even in real life. For example, a physician that always diagnoses an incoming patient as ill is a constant classifier. On the other hand, a degenerated – overpruned – decision tree (one leaf only) is a typical majority classifier.

In both cases all classifications are seemingly completely reliable. Obviously we also need to take in account the quality of classifier's underlying model and appropriately change our definition of reliability.

If we review our original definition of reliability (Eq. 1) it is immediately obvious that we assumed that the model was good. Our reliability estimations actually estimate the conditional reliability with respect to the model M

$$\mathrm{Rel}(y_i|M) = P(y_i \text{ is a true class of } x_i \mid \text{model M is good}) \tag{9}$$

To calculate required unconditional reliability we apply the conditional probability theorem for the whole model

$$\mathrm{Rel}'(y_i) = P(\text{model M is good}) * P(y_i \text{ is true class of } x_i \mid \text{model M is good}) \tag{10}$$

or even better for the partial models for each class y_i

$$\mathrm{Rel}'(y_i) = P(\text{model M is good for } y_i) * P(y_i \text{ is true class of } x_i \mid \text{model M is good for } y_i) \tag{11}$$

Now we only need to estimate the unconditional probabilities

$$P(\text{model is good}) \quad \text{or} \quad \forall i : P(\text{model is good for } y_i) \tag{12}$$

In Machine Learning we have many methods to estimate the quality of the induced model, e.g. a cross-validation computation of classification accuracy is suitable for estimation of Eq. 12. However it may be better to calculate it in a less coarse way, since at this point we already know the predicted class value (y_i).

We propose a calculation of (Bayesian) probability that the classification in a certain class is correct. Our approach is closely related to the calculation of post-test probabilities in medical diagnostics [3, 13]. Required factors can be easily estimated from the confusion matrix (Def. 1) with internal testing.

Definition 1. A *confusion matrix (CM)* is a matrix of classification errors obtained with an internal cross validation or leave-one-out testing on the training dataset. The ij-th element c_{ij} stands for the number of classifications to the class i that should belong to the class j.

Definition 2. Class sensitivity and specificity are a generalization of sensitivity (true positives ratio) and specificity (true negatives ratio) values for multi-class problems. Basically, for N classes we have N two-class problems. Let C_p be a correct class in certain case, and C a class, predicted by the classifier in the same case. For each of possible classes C_i, $i \in \{1..N\}$, we define its *class sensitivity* $Se(C_i) = P(C = C_i | C_p = C_i)$ and its *class specificity* $Sp(C_i) = P(C \neq C_i | C_p \neq C_i)$ as follows:

$$Se(C_i) = P(C = C_i | C_p = C_i) = \frac{c_{ii}}{\sum_j c_{ij}} \tag{13}$$

$$Sp(C_i) = P(C \neq C_i | C_p \neq C_i) = \frac{\sum_{j \neq i} c_{ji}}{\sum_{j \neq i} \sum_k c_{jk}} \tag{14}$$

Class conditional probability is calculated for each class C_i, given its prior probability $P(C_i)$, approximated with the prevalence of C_i in the training set, its class specificity (Sp) and sensitivity (Se):

$$\mathrm{P}_{\mathrm{cond}}(C_i) = \frac{P(C_i)\mathrm{Se}(C_i)}{P(C_i)\mathrm{Se}(C_i) + (1 - P(C_i))(1 - \mathrm{Sp}(C_i))} \tag{15}$$

For a fixed model and a fixed class C_i its class sensitivity and specificity are typically interdependent according to the ROC (receiver operating characteristics) curve (Fig. 3). An important advantage of class conditional probability over classification accuracy is that it takes in account both classifier's characteristics and prevalence of each class individually (Fig. 3). It is non-monotonic over all classes and therefore better describes the classifier's performance in its problem space.

To calculate the reliability estimation we therefore need the probability distributions P and Q, and index $i = \mathrm{argmax}\, P$ that determines the class with max. probability (C_i). According to the Eq. 11 we calculate the reliability estimations by

$$Rel(P, Q; C_i) = P_{\mathrm{cond}}(C_i) \times 2^{-J(P,Q)} \tag{16}$$

Multiplication by class conditional probabilities accounts for basic domain characteristics (prevalence of classes) as well as classifier's performance. This includes class sensitivity and specificity, and it is especially useful in an automatic setting for detecting possible anomalies such as default (either majority or constant classifiers) that – of course – cannot be trusted. It is easy to see that in this case we have one class with sensitivity 1 and specificity 0, whereas for all other classes we have sensitivity 0 and nonzero specificity. In the first case, the class post-test probability is equal to its prior probability, whereas in the second case it is 0.

3.4 Reliable and Unreliable Classifications

Since the datasets used for training classifiers vary in their representativeness and noise levels as well as Machine Learning algorithms vary in strength and assumptions of their underlying models, it is hard to obtain absolute thresholds for reliable classifications. In our experiments they varied between 0.20 and 0.70 for different domains and Machine Learning algorithms. Therefore it is useful to calibrate our criteria in advance by

utilizing the training dataset. On the training set, an internal cross validation or (better) leave-one-out testing is performed. For each training example a reliability estimation is made and the predicted as well as the exact class is known. In fact, we now have a new dataset with two possible classes {incorrectly-classified, correctly-classified}, and a single numeric attribute {reliability-estimation}. On this meta-problem we perform binary discretization of the reliability estimation attribute by maximizing the information gain of the split [4] with our goal being to obtain as pure subsets as possible. The best threshold T for the dataset split is calculated by maximizing Eq. 19.

$$H(S) = \text{entropy of the set } S \tag{17}$$

$$H(S;T) = \frac{S_1}{S}H(S_1) + \frac{S_2}{S}H(S_2) \quad \text{(entropy after split)} \tag{18}$$

$$\text{Gain}(S,T) = H(S) - H(S;T) \tag{19}$$

In the set S_1 there are unreliable examples $\{x : \text{Rel}(x) < T\}$ whereas in the set S_2 there are reliable examples $\{x : \text{Rel}(x) \geq T\}$. An experimental result for a dataset split is presented in Fig. 4. Note that internal testing must be done only once during the preparation for transductive reliability estimation. During this calculation we may also conveniently calculate necessary frequencies needed for model quality estimations (Def. 1).

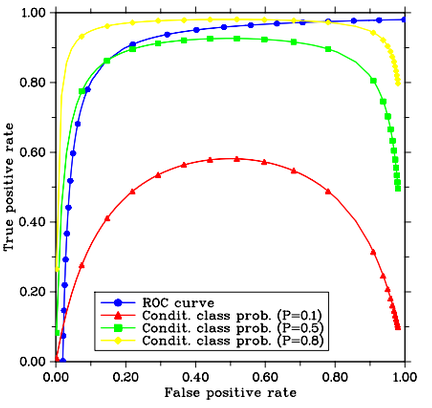


Fig. 3. Class conditional probabilities with respect to the ROC curve and the prior probability (P) of the class

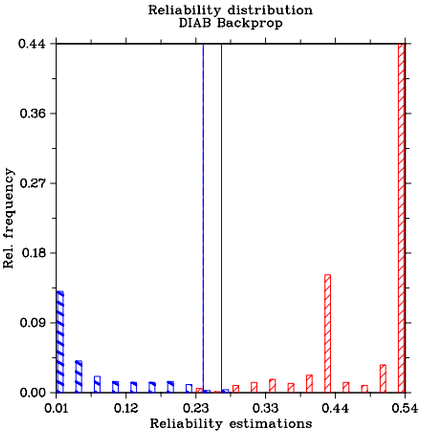


Fig. 4. Reliability estimations in domain "Diabetes" using Backpropagation neural networks. To the left of the possible two boundaries are unreliable classifications, to the right are the reliable classifications

4 Experiments

To validate our proposed methodology we performed extensive experiments with 6 different Machine Learning algorithms – naive and semi naive Bayesian classifier [7], backpropagation neural network [15], K-nearest neighbour, locally naive Bayesian classifier (a combination KNN and naive Bayesian classifier) [8], Assistant (ID3-like decision trees) [6] on 14 well-known benchmark datasets (Tab. 1a and 1b).

All algorithms were modified to represent their classifications as probability distributions. As a reference method the assigned classifier's probability was used. We performed two comparisons. Firstly, we tested how well can the original populations be split in the subpopulations of correctly and incorrectly classified examples. We applied Kolmogorov-Smirnov and χ^2 statistical tests. In all cases the difference between the two populations was significant with $p < 0.05$, in most cases even with $p \ll 0.01$. So

Table 1. Experimental results with transductive reliability estimation on 14 domains and 6 ML algorithms, obtained with leave one out testing

Domain	Inf. gain (Symm. K-L)	Inf. gain (class prob.)	Relative improvement	Kolmogorov-Smirnov test	χ^2-test
Mesh	0.32	0.18	87.97%	< 0.01	< 0.01
Breast cancer	0.14	0.06	142.76%	< 0.01	< 0.01
Nuclear	0.11	0.06	88.48%	< 0.01	< 0.01
Diabetes	0.23	0.09	195.44%	< 0.01	< 0.01
Heart	0.13	0.12	11.45%	< 0.01	< 0.01
Hepatitis	0.15	0.10	52.43%	< 0.01	< 0.01
Iris	0.18	0.15	33.98%	< 0.01	< 0.01
Chess endgame	0.07	0.04	145.28%	< 0.01	< 0.01
LED	0.08	0.06	10.93%	< 0.01	< 0.01
Lymphograpy	0.13	0.10	30.66%	< 0.01	< 0.01
Primary tumor	0.22	0.13	78.54%	< 0.01	< 0.01
Rheumatology	0.29	0.15	105.28%	< 0.01	< 0.01
Soybean	0.17	0.11	83.05%	< 0.01	< 0.01
Voting	0.11	0.09	20.31%	< 0.01	< 0.01

(a) Average results on different domains

ML algorithm	Inf. gain (Symm. K-L)	Inf. gain (class prob.)	Relative improvement	Kolmogorov-Smirnov test	χ^2-test
Naive Bayes	0.18	0.11	82.31%	< 0.01	< 0.01
Semi naive Bayes	0.16	0.10	56.31%	< 0.01	< 0.01
Neural network	0.20	0.08	169.38%	< 0.01	< 0.05
K-nearest neighbour	0.13	0.09	55.19%	< 0.05	< 0.01
KNN + Naive Bayes	0.16	0.12	43.10%	< 0.01	< 0.01
Assistant	0.15	0.11	32.26%	< 0.01	< 0.01

(b) Average results of different Machine Learning algorithms

the splitting criterion introduced in Sec. 3.4 really produces statistically significantly different subpopulations.

Secondly, we measured the improvement of our methodology over the assigned classifier's probability. For both methods we compared information gains (Sec. 3.4) that directly correspond to the (im)purity of the split subpopulations. Results are summarized by domains (Tab. 1a) and Machine Learning algorithms (Tab. 1b).

As it is clearly visible from the results, relative improvements were always in favour of transductive reliability estimation. After the split, the subpopulations were much purer than the original one, information gain (Eq. 19) was on average increased by 75%, ranging between 11% and 195%. All improvements were statistically significant using a two-tailed t-test with $p < 0.05$.

We also performed an in-depth comparison of transductive reliability estimations and physicians' reliability estimations in the *nuclear* dataset (nuclear diagnostics of Coronary Artery Disease), where expert physicians were available for cooperation[10]. Our method increased the number of correctly reliable classifications by 22.5% while the number of incorrectly marked as reliable classifications remaind the same [9]. It is estimated that such results if applicable in practice would reduce the costs of diagnostic process by 10%!

5 Discussion

We propose a new methodology for transductive reliability estimations of classifications within Machine Learning framework. We provide a theoretical framework for our methodology and an efficient implementation in conjunction with any Machine Learning algorithm that can represent its predictions as probability distributions. We show that in certain extreme cases our basic approach fails and provide improvements that account for such anomalous cases. We argue that, especially in risk-sensitive applications, any serious Machine Learning tool should use a similar methodology for the assessment single of classification reliability. Another use of reliability estimations is in combining answers from different predictors, weighed according to their reliability.

Our experiments in benchmark domains show that our approach is significantly better than evaluating classifier's posterior probabilities. Experimental results of reliability estimations in the Coronary Artery Disease diagnostics also show enormous potential of our methodology. The potential improvements in diagnostic process are so big that the physicians are seriously considering introducing this approach in everyday diagnostic practice.

There are several things that can be done to further develop our approach. Currently we aim to replace the discretization of reliability estimation values for obtaining a threshold value. We intend to replace it with proprietary population statistics that would hopefully eliminate impact of differently representative datasets and model weaknesses on resulting quantitative reliability estimation values.

Acknowledgements

We thank dr. Ciril Grošelj, from the Nuclear Medicine Department, University Medical Centre Ljubljana, for his work while collecting the *nuclear* data and interpreting the results, and the anonymous reviewers for their insightful comments. This work was supported by the Slovenian Ministry of Education, Science and Sports.

References

[1] S. D. Bay and M. J. Pazzani. Characterizing model errors and differences. In *Proc. 17th International Conf. on Machine Learning*, pages 49–56. Morgan Kaufmann, San Francisco, CA, 2000. 220

[2] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 92–100, 1998. 220

[3] G. A. Diamond and J. S. Forester. Analysis of probability as an aid in the clinical diagnosis of coronary artery disease. *New England Journal of Medicine*, 300:1350, 1979. 219, 225

[4] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In *Proc. ICML'95*, pages 194–202. Morgan Kaufmann, 1995. 227

[5] A. Gammerman, V. Vovk, and V. Vapnik. Learning by transduction. In *Proceedings of the* 14^{th} *Conference on Uncertainty in Artificial Intelligence*, pages 148–155, Madison, Wisconsin, 1998. 221, 222

[6] I. Kononenko, E. Šimec, and M. Robnik-Šikonja. Overcoming the myopia of inductive learning algorithms with ReliefF. *Applied Intelligence*, 7:39–55, 1997. 228

[7] I. Kononenko. Semi-naive Bayesian classifier. In Y. Kodratoff, editor, *Proc. European Working Session on Learning-91*, pages 206–219, Porto, Potrugal, 1991. Springer-Verlag. 228

[8] M. Kukar. *Estimating classifications' reliability*. PhD thesis, University of Ljubljana, Faculty of Computer and Information Science, Ljubljana, Slovenia, 2001. In Slovene. 220, 225, 228

[9] M. Kukar. Making reliable diagnoses with machine learning: A case study. In Silvana Quaglini, Pedro Barahona, and Steen Andreassen, editors, *Proceedings of Artificial Intelligence in Medicine Europe, AIME 2001*, pages 88–96, Cascais, Portugal, 2001. Springer. 221, 229

[10] M. Kukar, I. Kononenko, C. Grošelj, K. Kralj, and J. Fettich. Analysing and improving the diagnosis of ischaemic heart disease with machine learning. *Artificial Intelligence in Medicine*, 16 (1):25–50, 1999. 229

[11] M. Li and P. Vitányi. *An introduction to Kolmogorov complexity and its applications*. Springer-Verlag, New York, 2^{nd} edition, 1997. 221, 222

[12] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39 (2/3):103–134, 2000. 220

[13] M. Olona-Cabases. The probability of a correct diagnosis. In J. Candell-Riera and D. Ortega-Alcalde, editors, *Nuclear Cardiology in Everyday Practice*, pages 348–357. Kluwer, 1994. 225

[14] J. Ortega, M. Koppel, and S. Argamon. Arbitrating among competing classifiers using learned referees. *Knowledge and Information Systems Journal*, 3:470–490, 2001. 220

[15] D. E. Rumelhart and J. L. McClelland. *Parallel Distributed Processing*, volume 1: Foundations. MIT Press, Cambridge, 1986. 228

[16] C. Saunders, A. Gammerman, and V. Vovk. Transduction with confidence and credibility. In *Proceedings of the International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, 1999. 219, 221, 222

[17] A. Seewald and J. Furnkranz. An evaluation of grading classifiers. In *Proc. 4th International Symposium on Advances in Intelligent Data Analysis*, pages 115–124, 2001. 220

[18] I. J. Taneja. On generalized information measures and their applications. *Adv. Electron. and Elect. Physics*, 76:327–416, 1995. 223

[19] K. M. Ting. Decision combination based on the characterisation of predictive accuracy. *Intelligent Data Analysis*, 1:181–206, 1997. 219

[20] V. Vapnik. *Statistical Learning Theory*. John Wiley, 1998. 220

[21] V. Vovk, A. Gammerman, and C. Saunders. Machine learning application of algorithmic randomness. In *Proceedings of the* 16^{th} *International Conference on Machine Learning (ICML'99)*, Bled, Slovenija, 1999. 219, 221, 222