Knowl Inf Syst (2006) 9(3): 364–384
DOI 10.1007/s10115-005-0203-z

Knowledge and Information Systems

REGULAR PAPER

Matjaž Kukar

Quality assessment of individual classifications in machine learning and data mining

Received: 1 November 2004 / Revised: 15 January 2005 / Accepted: 19 February 2005 /
Published online: 9 September 2005

Abstract Although in the past machine learning algorithms have been successfully used in many problems, their serious practical use is affected by the fact that often they cannot produce reliable and unbiased assessments of their predictions' quality. In last few years, several approaches for estimating reliability or confidence of individual classifiers have emerged, many of them building upon the algorithmic theory of randomness, such as (historically ordered) transduction-based confidence estimation, typicalness-based confidence estimation, and transductive reliability estimation. Unfortunately, they all have weaknesses: either they are tightly bound with particular learning algorithms, or the interpretation of reliability estimations is not always consistent with statistical confidence levels. In the paper we describe typicalness and transductive reliability estimation frameworks and propose a joint approach that compensates the above-mentioned weaknesses by integrating typicalness-based confidence estimation and transductive reliability estimation into a joint confidence machine. The resulting confidence machine produces confidence values in the statistical sense. We perform series of tests with several different machine learning algorithms in several problem domains. We compare our results with that of a proprietary method as well as with kernel density estimation. We show that the proposed method performs as well as proprietary methods and significantly outperforms density estimation methods.

Keywords Data mining · Machine learning · Typicalness · Transduction · Quality assessment

M. Kukar (✉)
Faculty of Computer and Information Science, University of Ljubljana, Tržaška 25,
SI-1001 Ljubljana, Slovenia
E-mail: matjaz.kukar@fri.uni-lj.si

1 Introduction

Usually machine learning algorithms output only bare predictions (classifications) for the new unclassified examples. While there are ways for almost all machine learning algorithms to at least partially provide quantitative assessment of the particular classification, so far there is no general method to assess the quality (confidence, reliability) of a single classification. We are interested in the assessment of classifier's performance on a *single example* and not in average performance on an independent dataset. 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 an overall quality measure of a classifier (e.g. classification accuracy, mean squared error, . . .) with respect to the whole input distribution would not provide the desired value. Another possible use of quality assessment of single classifications is in ensembles of machine learning algorithms for selecting or combining answers from different classifiers [13].

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 confidence (reliability) of its prediction. However, such estimations may in general not be good due to inherent biases of the applied algorithms.[1] Reliability estimation of a classification ($\widetilde{y}$) of a single example (x), given its true class (y) should have the following property:

$$\mathrm{Rel}(\widetilde{y} \mid x) = t \;\Rightarrow\; P(\widetilde{y} \neq y) \leq 1 - t \tag{1}$$

If Eq. (1) holds, or even better, if it approaches equality, a reliability measure can be treated as a confidence value [15].

1.1 Related work

Several methods for inducing probabilistic descriptions from training data, figuring the use of density estimation algorithms, are emerging as an alternative to more established approaches for machine learning. Frequently kernel density estimation [30] is used for density estimation of input data using diverse machine learning paradigms such as probabilistic neural networks [25], Bayesian networks and classifiers [9], and decision trees [24]. By this approach a chosen paradigm, coupled with kernel density estimation, is used for modelling the probability distribution of input data. Alternatively, stochastically changing class labels in the training dataset is proposed [6] in order to estimate conditionally class probability.

There is some ongoing work for constructing classifiers that divide the data space into reliable and unreliable regions [1]. Such meta-learning approaches have

[1] An extreme case of inherent bias can be found in a trivial constant classifier that blindly labels any example with a predetermined class with self-proclaimed confidence 1.

also been used for picking the most reliable prediction from the outputs of an ensemble of classifiers [23].

Meta learning community is partially dealing with predicting the right machine learning algorithm for a particular problem [18] based on performance and characteristics of other, simpler learning algorithms. In our problem of confidence estimation such an approach would result in learning to predict confidence value based on characteristics of single examples.

Much work has been done in applications of the transduction methodology [22], in connection with algorithmic theory of randomness. Here, approximations of randomness deficiency for different methods (SVMs, ridge regression) have been constructed in order to estimate confidence of single predictions. The drawback of this approach is that confidence estimations need to be specifically designed for each particular method and cannot be applied to other methods.

Another approach to reliability estimation, similarly based on the transduction principle, has been proposed in [13]. While it is general and independent of the underlying classifier, interpretation of its results isn't always possible in the statistical sense of confidence levels.

A few years ago typicalness has emerged as a complementary approach to transduction [8, 15, 19]. By this approach, a "strangeness" measure of a single example is used to calculate its typicalness, and consequently a confidence in classifier's prediction. The main drawback of this approach is that for each machine learning algorithm an appropriately constructed strangeness measure is needed.

In this paper we present a further development of the latter two approaches where transductive reliability estimation serves as a generic strangeness measure in the typicalness framework. We compare the experimental results to those of kernel density estimation and show that the proposed method significantly outperforms it. We also suggest how the basic transduction principle can be used to significantly improve the results of kernel density estimation so it almost achieves the results of transductive typicalness.

The paper is organized as follows. In Sect. 2 we describe the basic ideas of typicalness and transduction, outline the process of their integration, and review kernel density estimation methods used for comparison. In Sect. 3 we evaluate how our methodology compares to other approaches in 15 domains with 6 machine learning algorithms. In Sect. 4 we present some conclusions and directions for future work.

2 Methods and materials

The produced confidence values should be valid in the following sense. Given some possible label space $\mathcal{Y}$, if an algorithm predicts some set of labels $Y \subseteq \mathcal{Y}$ with confidence t for a new example which is truly labelled by $y \in \mathcal{Y}$, then we would expect the following to hold over randomization of the training set and the new example:

$$P(y \notin Y) \leq 1 - t \tag{2}$$

Note that Eq. (2) is very general and valid for both classification (Y is predicted set of classes) and regression problems (Y is a predicted interval). As we deal only with single predictions in this paper, Eq. (2) can be simplified to a single predicted class value ($Y = \{\widetilde{y}\}$):

$$P(y \neq \widetilde{y}) \leq 1 - t \tag{3}$$

2.1 Typicalness

In the typicalness framework [15, 16, 22] we consider a sequence of examples $(z_1, \ldots, z_n) = ((x_1, y_1), \ldots, (x_n, y_n))$, together with a new example x_{n+1} with unknown label $\widetilde{y}_{n+1}$, all drawn independently from the same distribution over $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ where $\mathcal{X}$ is an attribute space and $\mathcal{Y}$ is a label space. Our only assumption is therefore that the training as well as new (unlabelled) examples are independently and identically distributed (*iid* assumption).

We can use the typicalness framework to gain confidence information for each possible labelling for a new example x_{n+1}. We postulate some labels $\widetilde{y}_{n+1}$ and for each one we examine how likely (typical) it is that all elements of the extended sequence $((x_1, y_1), \ldots, (x_{n+1}, \widetilde{y}_{n+1}))$ might have been drawn independently from the same distribution or how typically *iid* the sequence is. The more typical the sequence, the more confident we are in $\widetilde{y}_{n+1}$. To measure the typicalness of sequences, we define, for every $n \in \mathbb{N}$, a typicalness function $t : \mathcal{Z}^n \rightarrow [0, 1]$ which, for any $r \in [0, 1]$ has the property

$$P((z_1, \ldots, z_n) \ : \ t(z_1, \ldots, z_n) \leq r) \leq r \tag{4}$$

If a typicalness function returns 0.05 for a given sequence, we know that the sequence is unusual because it will be produced at most 5% of the time by any *iid* process. It has been shown [15] that we can construct such functions by considering the "strangeness" of individual examples. If we have some family of functions

$$f : \mathcal{Z}^n \times \{1, 2, \ldots, n\} \rightarrow \mathbb{R}, \ n \in \mathbb{N}, \ldots, \tag{5}$$

then we can associate a strangeness value

$$\alpha(z_i) = f(\{z_1, \ldots, z_n\}; i), \quad i = 1, 2, \ldots, n \tag{6}$$

with each example and define the following typicalness function

$$t((z_1, \ldots, z_n)) = \frac{\#\{\alpha(z_i) : \alpha(z_i) \geq \alpha(z_n)\}}{n} \tag{7}$$

We group individual strangeness functions α_i into a family of functions $A_n : n \in \mathbb{N}$, where $A_n : \mathcal{Z}^n \rightarrow \mathbb{R}^n$ for all n. This is called an individual strangeness measure if, for any n, any permutation $\pi : \{1, \ldots, n\} \rightarrow \{1, \ldots, n\}$, any sequence $(z_1, \ldots, z_n) \in \mathcal{Z}^n$, and any $(\alpha_{\pi(1)}, \ldots, \alpha_{\pi(n)}) \in \mathbb{R}^n$) it satisfies the following criterion [15]:

$$(\alpha_1, \ldots, \alpha_n) = A_n(z_1, \ldots, z_n) \Longrightarrow \big(\alpha_{\pi(1)}, \ldots, \alpha_{\pi(n)}\big) = A_n\big(z_{\pi(1)}, \ldots, z_{\pi(n)}\big) \tag{8}$$

The meaning of this criterion is that the same value should be produced for each individual element in sequence, regardless of the order in which their individual strangeness values are calculated. This is a very important criterion, because it can be proven [15] that the constructed typicalness function Eq. (7) satisfies the condition from Eq. (4), provided that the individual strangeness measure satisfies the Eq. (8).

From a practical point of view it is advisable [15] to use positive strangeness measures, ranging between 0 for most typical examples, and some positive upper bound, (up to $+\infty$), for most atypical examples.

2.1.1 Typicalness in machine learning

In the machine learning setup, for calculating the typicalness of a new example $z_{n+1} = (x_{n+1}, \widetilde{y}_{n+1})$ described with attribute values x_{n+1} and labelled $\widetilde{y}_{n+1}$, given the training set $(z_1, \ldots, z_n)$, Eq. (7) changes to

$$t((z_1, \ldots, z_{n+1})) = \frac{\#\{\alpha(z_i) : \alpha(z_i) \geq \alpha(z_{n+1})\}}{n+1} \tag{9}$$

Note that on the right-hand side of Eq. (9), z_i belongs to the extended sequence, i.e. $z_i \in \{z_1, \ldots, z_{n+1}\}$. For a given machine learning algorithm, first we need to construct an appropriate strangeness measure and modify the algorithm accordingly.[2] Then, for each new unlabelled example x, all possible labels $\widetilde{y} \in Y$ are considered. For each label $\widetilde{y}$ a typicalness of labelled example $t((x, \widetilde{y})) = t((z_1, \ldots, z_n, (x, \widetilde{y})))$ is calculated. Finally, the example is labelled with "most typical" class, that is the one that maximizes $\{t((x, \widetilde{y}))\}$. By Eq. (7) the second largest typicalness is an upper bound on the probability that the excluded classifications are correct [19]. Consequently, the confidence is calculated as follows:

$$\text{confidence}((x, \widetilde{y})) = 1 - \text{typicalness of second most typical label.} \tag{10}$$

2.2 Transductive 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 has also been advocated in the context of co-training. It has been shown [2] that for every better-than-random classifier its performance can be significantly boosted by utilizing only additional unlabelled data.

It has been suggested [27] 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).

[2] This is the main problem of the typicalness approach, as the algorithms need do be considerably changed.

2.2.1 A formal background

Let $\mathcal{X}$ be a space of attribute descriptions of points (examples) in a training sample (dataset), 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

$$S = \{(x_1, y_1), \ldots, (x_l, y_l)\} \tag{11}$$

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

$$W = \{x_{l+1}, \ldots, x_{l+m}\} \tag{12}$$

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 $S \cup W$). In contrast, inductive inference aims at choosing a single function $f \in \mathcal{H}$ that is best suited to the unknown probability distribution $\mathcal{P}$.

At this point there arises a question how to calculate labels of points from 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 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. In case of $m = 1$ we can significantly reduce the computational complexity by labelling a point with a label predicted from S only [12].

By leveraging the *iid* sampling assumption and transductive inference, one can for each labelling estimate its reliability (also referred to as confidence, a probability that it is correct). If the *iid* 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 in general belongs to the non-computable class [14], so approximation methods have to be used [12, 29].

Evaluation of the prediction reliability for single points in a data space has many uses. In risk-sensitive applications (medical diagnosis, financial and critical control applications) it often matters, how much one can rely upon a given prediction. In such a case a general reliability measure of a classifier (e.g. classification accuracy, mean, squared error, ...) with respect to the whole input distribution would not provide the desired warranty. Another use of reliability estimations is in combining answers from different predictors, weighed according to their reliability.

2.2.2 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, both measured in bits. We define the *randomness deficiency* of the string v as following [14, 29]:

$$\delta(v) = l(v) - C(v) \tag{13}$$

Randomness deficiency measures how random is the respective binary string and therefore the set it represents. The larger it is, more regular is 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 with largest randomness deficiency as the most probable one [29]. That is, we would select the most regular one. We could also construct a universal Martin-Löf's test for randomness [14]:

$$\sum\{P(x \mid l(x) = n) : \delta(x) \geq m\} \leq 2^{-m} \tag{14}$$

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 [29].

Unfortunately, as the definition of randomness deficiency is based on the Kolmogorov complexity, it is not computable. Therefore we need feasible approximations to use this principle in practice. Extensive work has been done by using Support Vector Machines [4, 22, 29], however no general approach exists so far.

2.2.3 A machine learning interpretation

In machine learning terms, the sets S and $S \cup W$ are represented by the induced models M_S and $M_{S \cup W}$. The randomness of the sets is reflected in the (Kolmogorov) complexity of the respective models. If for the set $S \cup W$ the labelling of W with the largest randomness deficiency is selected, it follows from our definition of randomness deficiency (Eq. (13)) that since the length $l(v)$ is constant, the Kolmogorov complexity $C(M_{S \cup W})$ is minimal. Therefore the model $M_{S \cup W}$ is most similar to the M_S.

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 between them 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 the atypical 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 this example. 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.

The transductive reliability estimation process and its theoretical foundations originating from Kolmogorov complexity are described in more detail in [13]. Basically, we have a two-step process, featuring an *inductive step* followed by a *transductive step*.

- 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. An example, labelled with the classified class is temporarily included into the training set.
- 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 and classified using the transduced classifier. Both classifications of the same example are compared and their difference (distance) is calculated, thus approximating the randomness deficiency.
- After the reliability is calculated, the example in question is removed from the training set.

In practice the inductive step is performed only once, namely on the original training set. New examples are not permanently included in the training set; this would be improper since the correct class is at this point still unknown. Although retraining for each new example seems to be highly time consuming, it is not such a problem in practice, especially if incremental learners (such as naive Bayesian classifier) are used.

A brief algorithmic sketch is given in Fig. 1. An intuitive explanation of transductive reliability estimation is that we disturb a classifier by inserting a new example in a training set. A magnitude of this disturbance is an estimation of the classifier's instability (unreliability) in a given region of its problem space.

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 measure D should ideally satisfy all requirements for a distance (i.e. nonnegativity, triangle nonequality and symmetry), however in practice nonnegativity suffices. For calculating the difference between probability distributions, a *Kullback-Leibler*

```
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
     · 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
     · classify this example with a transduced classifier
3: Calculate a randomness deficiency approximation as a normalized difference
   $J_N(P,Q)$ between inductive ($P$) and transductive ($Q$) classification.
4: Calculate the reliability of classification as in a universal Martin-Löf's test for ran-
   domness 1-normalized difference
```

Fig. 1 The algorithm for transductive reliability estimation

divergence is frequently used [5, 26]. A Kullback-Leibler divergence, sometimes referred to as a relative entropy or I-divergence, is defined between probability distributions P and Q:

$$I(P, Q) = -\sum_{i=1}^{n} p_i \log_2 \frac{p_i}{q_i} \tag{15}$$

In our experiments we use a symmetric Kullback-Leibler divergence, or J-divergence, which is defined as follows:

$$J(P, Q) = (I(P, Q) + I(Q, P)) = \sum_{i=1}^{n} (p_i - q_i) \log_2 \frac{p_i}{q_i} \tag{16}$$

$J(P, Q)$ is limited to the interval $[0, \infty]$, where $J(P, P) = 0$. Since in this context we require the values to be from the $[0, 1]$ interval we normalize the divergence in the spirit of Martin-Löf's test for randomness.

$$J_N(P, Q) = 1 - 2^{-J(P,Q)} \tag{17}$$

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

2.2.4 Assessing the classifier's quality: the curse of trivial models

So far we have implicitly assumed that the model produced 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 [11].

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.

Obviously we assume that the learnt (induced) data model is good. Our reliability estimations actually estimate the conditional reliability with respect to the model M

$$\mathrm{Rel}(y_i \mid M) = P(y_i \text{ is a true class of } x_i \mid \text{model M is good}) \tag{18}$$

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

$$\mathrm{Rel}'(y_i) = P(\text{model M is good}) * P(y_i \text{ is true class of } x_i \mid \text{model M is good}) \tag{19}$$

or even better for the partial models for each class y_i

$$\begin{aligned}\mathrm{Rel}'(y_i) = {} & P(\text{model M is good for } y_i) \\ & *P(y_i \text{ is true class of } x_i \mid \text{model M is good for } y_i)\end{aligned} \tag{20}$$

Now we only need to estimate the unconditional probabilities

$$P(\text{model is good}) \quad \text{or} \quad \forall i : P(\text{model is good for } y_i) \tag{21}$$

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. (21). 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 (Bayesian) calculation of 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, 17]. Required factors can be easily estimated from the confusion matrix (Definition 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.

$$CM = \begin{pmatrix} c_{11} & c_{12} & c_{13} & \dots & c_{1N} \\ c_{21} & c_{22} & c_{23} & \dots & c_{2N} \\ c_{31} & c_{32} & c_{33} & \dots & c_{3N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{N1} & c_{N2} & c_{N3} & \dots & c_{NN} \end{pmatrix}$$

$$c_{ij} = \text{number of classifications to class i that belong to class } j \tag{22}$$

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*

$Se(C_i) = P(C = C_i \mid C_p = C_i)$ and its *class specificity* $Sp(C_i) = P(C \neq C_i \mid C_p \neq C_i)$ as follows:

$$Se(C_i) = P(C = C_i \mid C_p = C_i) = \frac{c_{ii}}{\sum_j c_{ij}} \tag{23}$$

$$Sp(C_i) = P(C \neq C_i \mid C_p \neq C_i) = \frac{\sum_{j \neq i} c_{ji}}{\sum_{j \neq i} \sum_k c_{jk}} \tag{24}$$

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*):

$$\mathrm{P}_{\mathrm{cond}}(C_i) = \frac{P(C_i)\mathrm{Se}(C_i)}{P(C_i)\mathrm{Se}(C_i) + (1 - P(C_i))(1 - \mathrm{Sp}(C_i))} \tag{25}$$

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. (20) we calculate the reliability estimations by

$$Rel(P, Q; C_i) = \mathrm{P}_{\mathrm{cond}}(C_i) \times J_N(P, Q) \tag{26}$$

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.

2.3 Merging the typicalness and transduction frameworks

There is a very good reason for merging typicalness and transductive reliability estimation frameworks together. While transduction gives good reliability estimations, they are often hard to interpret in the statistical sense. On the other hand, the typicalness framework gives clear confidence values, however in order to achieve this a good strangeness measure $\alpha(z_i)$ needs to be constructed.

Of course, there is a trivial solution to it, namely a uniform strangeness measure $\alpha_i = C$, where C is some constant value. Unfortunately, this does us no good, since it treats all examples as equally strange and can be considered as most conservative strangeness measure. It is therefore necessary to construct a sensible strangeness measure. In [15, 19] some ideas on how to construct strangeness measures for different machine learning algorithms are presented.

On the other hand, as we shall see later, for a strangeness measure we can always use transductive reliability estimation. We may speculate that most reliable examples are also least strange. Therefore we define the strangeness measure for a new example $z_{n+1} = (x_{n+1}, \widetilde{y}_{n+1})$, described with attribute values x_{n+1} and labelled $\widetilde{y}_{n+1}$, given the training set $(z_1, \ldots, z_n)$ as follows:

$$\alpha(z_{n+1}) = f(z_1, \ldots, z_{n+1}; n+1) = 1 - \mathrm{Rel}(z_{n+1}) \in [0, 1] \tag{27}$$

It can be shown that such a strangeness function satisfies the criterion from Eq. (8) and therefore has the property required by Eq. (7).

Theorem 1 *The strangeness measure $\alpha(z_i) = 1 - \mathrm{Rel}((x_i, \widetilde{y}_i))$ is independent of the order in which the examples' strangeness values are calculated.*

Proof The training set is only temporarily changed by including a suitably labelled new example in a transductive step (Fig. 1. It is restored back to the initial training set as soon as the reliability estimation is calculated. Therefore the training set remains invariant for all new examples for which the reliability estimation needs to be calculated. It follows that it is irrelevant in which order the examples are presented and the criterion for Eq. (8) is therefore satisfied. Note that Eq. (8) does not require that examples are ordered in any particular way, but only that any permutation of the order of their evaluations produces the same result for each example. □

Consequently we can, for any machine learning classifier, universally use a strangeness measure $\alpha((x, \widetilde{y})) = 1 - Rel((x, \widetilde{y}))$ (although, as we shall see later, in the typicalness setting this expression can be even more simplified). It is positive, and the "more strange" examples have higher strangeness values, as suggested in [15].

2.3.1 Simplification of transductive reliability estimation for application within the typicalness framework

Alternatively, the calculation of the strangeness measure can, in the context of typicalness and reliability estimation, be much simplified. Simplifications are twofold.

1. Since the only requirement for strangeness measure is that is is positive, no transformations to [0, 1] interval are necessary. The transformation is actually performed by Eq. (9).
2. The typicalness framework efficiently deals with extremely deviant classifiers (such as those from Sect. 2.2.4). As an example, let us consider the most "pathological" case, the constant classifier. With constant classifiers, all strangeness values are equal (i.e. all examples are equally – maximally – strange). Note that in this case magnitudes of strangeness values are irrelevant as they are all the same. By Eq. (9) it follows that for all possible classifications of every (new) example the typicalness is therefore 1.0. By Eq. (10) this yields confidence of 0. Such trivial classifiers are therefore maximally distrusted.

Let $P_{(x,\widetilde{y})}$ and $Q_{(x,\widetilde{y})}$ be the probability distributions obtained after the inductive step ($P_{(x,\widetilde{y})}$) and transductive step ($Q_{(x,\widetilde{y})}$) of the algorithm from Fig. 1. It can easily be shown that Theorem 1 holds also for $\alpha((x, \widetilde{y})) = J(P_{(x,\widetilde{y})}, Q_{(x,\widetilde{y})})$ (symmetric Kullback-Leibler divergence) as well as for $\alpha((x, \widetilde{y})) = I(P_{(x,\widetilde{y})}, Q_{(x,\widetilde{y})})$ (asymmetric Kullback-Leibler divergence).

Implementing a transductive reliability estimation in a typicalness framework is straightforward. For all training examples, reliability estimation is calculated by leave-one-out testing, and they are labelled as correctly or incorrectly classified. For each new example x with classification $\widetilde{y}$ its confidence $\text{conf}((x,\widetilde{y}))$ is calculated as in Sect. 2.1, Eq. (10). Regardless of the number of classes in original problem, there are only two possibilities (meta-classes) for each classification. It is either correct or incorrect. Therefore we always deal with exactly two meta-classes that represent correct classifications and incorrect classifications. As we want the confidence to reflect the probability of a correct classification, we need to invert the confidence values for incorrect meta-class:

$$\text{confidence}((x,\widetilde{y})) = \begin{cases} \text{conf}((x,\widetilde{y})) & \text{“correct” meta-class,} \\ 1 - \text{conf}((x,\widetilde{y})) & \text{“incorrect” meta-class.} \end{cases} \tag{28}$$

2.4 Meta-learning and kernel density estimation

The problem of estimating a confidence value can also be viewed as a meta-learning problem where the original class value is replaced by the correctness of its prediction. Let $\widehat{y}$ be a meta-class for training examples obtained with internal leave-one-out testing (i.e. $\widehat{y} = 1$ for correct and $\widehat{y} = 0$ for incorrect classifications). We can calculate the confidence in a given prediction of a new, previously unseen example x by estimating the function $\widehat{y}(x)$ with a nearest neighbour classifier:

$$\widehat{y}(x) = \frac{1}{K} \sum_{x_i \in N_K(x)} \widehat{y}_i(x_i) \tag{29}$$

Here $N_K(x)$ is the set of K points nearest to x according to some distance measure. However, such simple estimations may be problematic when the attribute space is large (lots of multi-valued, possibly correlated, attributes), and sparsely populated (relatively small number of training examples). Our experimental results (Table 2) also shows this problem, as using a nearest neighbour meta-learner results in lowest performance of all methods.[3] Therefore, a transformation of input space is necessary to reduce the dimensionality of input space. We have chosen the principal component analysis (PCA) methodology on the training data, and two components with largest variances were selected as data descriptors. On average, the sum of the two components' relative variances is about 0.7. This means, that the two principal components describe about 70% of data variability.

Rather than giving the nearest neighbours equal weights, we can assign them weights that decrease smoothly with distance from the target point. This leads us to kernel density estimation [28] in reduced and uncorrelated data space. It can be estimated by using the Nadaraya-Watson kernel weighted average:

$$\widehat{y}(x) = \frac{\sum_{i=i}^{N} K_\lambda(x,x_i)\widehat{y}_i(x_i)}{\sum_{i=i}^{N} K_\lambda(x,x_i)} \tag{30}$$

[3] To be fair, it must be said that other more advanced meta-learners could have been used. However, this was not the aim of the paper.

where $\lambda = [\lambda_1, \lambda_2]$ is a vector of kernel parameters (bandwidths), and $K_\lambda(x, x_i)$ is a simplified (uncorrelated) bivariate gaussian kernel:

$$K_{\lambda_1,\lambda_2}(x, x_i) = \frac{1}{2\pi\lambda_1\lambda_2} e^{-\frac{1}{2}\frac{(x[1]-x_i[1])^2}{\lambda_1^2} + \frac{(x[2]-x_i[2])^2}{\lambda_1^2}} \tag{31}$$

As the PCA involves a numerical procedure that transforms a number of possibly correlated input variables (attributes) into a (smaller) number of uncorrelated variables (principal components), it is therefore perfectly justified to use a simplified bivariate Gaussian kernel for density estimation on uncorrelated variables. Our experiments have shown, that indeed in all cases the correlation between the largest two principal components is less than 10^{-14}, also negligible. For the bivariate Gaussian kernels, appropriate bandwidths were calculated from training data according to the rule of thumb as described by Wand [30, p. 98].

For each dataset and algorithm the following procedure was performed. For each training example, a correctness of its classification was determined by the leave-one-out testing methodology. Training examples were partitioned in sets of correctly and incorrectly classified examples, and used for kernel density estimations of correct and incorrect classifications. For each new examples, principal components were calculated and used to calculate the density of correct classifications (cd) as well as the density of incorrect classifications (id) at respective coordinates. The confidence value of a new example was calculated as $cd/(cd + id)$ [7].

2.5 Improving kernel density estimation by transduction principle

The procedure described in Sect. 2.4 is computationally fast when applied to new examples as it involves only calculating the principal components (scaling and one matrix multiplication), and two fast uncorrelated density estimations. Unfortunately, its performance (Table 2) compared to transductive confidence estimation is rather uninspiring. The performance, however, can be easily improved by using some ideas from meta learning and transduction frameworks. Namely, we can easily extend the original data description by including the predicted class as well as class probability distributions. They may be obtained with internal leave-one-out testing on the training set.

On extended data the principal components are calculated. A new example's class and class distribution is predicted by the original classifier, and the example's description is enhanced by the classifier's prediction. An enhanced example description is then used in the density estimation procedure as described in Sect. 2.4.

2.6 Testing methodology

To validate the proposed methodology we performed extensive experiments with 6 different machine learning algorithms – naive and semi naive Bayesian classifier [10], backpropagation neural network [21], K-nearest neighbour, locally naive Bayesian classifier (a combination of KNN and naive Bayesian classifier)

[13], two kinds of Assistant (ID3-like decision trees) with both information gain and ReliefF [20] impurity measures. Experiments were performed with 14 well-known benchmark datasets from the UCI repository (Mesh, Breast cancer, Diabetes, Heart, Hepatitis, Iris, Chess endgame (king-rook vs. king), LED, Lymphography, Primary tumor, Rheumatology, Soybean, Voting), and on a real-life problem of nuclear cardiology diagnostics (Nuclear).

For each dataset and algorithm we determined for each training example by internal leave-one-out testing its correctness – whether it was correctly (1) or incorrectly (0) classified. For reliability estimations, confidence values and density estimations, we calculated their correlation with correctness. In an ideal case (each correct example has value 1, each incorrect 0), the result would be 1.

We also measured how well a method discriminates between correctly and incorrectly classified examples. For each method (reliability estimations, confidence values, and density estimations) we calculated the boundary b that maximizes purity (information gain) of the discriminated examples. The boundary b is calculated by maximizing Eq. (33).

$$\begin{aligned} H(S) &= -\frac{|S_1|}{|S|}\log_2\frac{|S_1|}{|S|} - \frac{|S_2|}{|S|}\log_2\frac{|S_2|}{|S|} \quad \text{(entropy before split)} \\ H(S;b) &= \frac{|S_1|}{|S|}H(S_1) + \frac{|S_2|}{|S|}H(S_2) \quad \text{(entropy after split)} \\ \mathrm{Gain}(S,b) &= H(S) - H(S;b) \end{aligned} \tag{32}$$

Here, S is the set consisting of all examples, in the set S_1 there are unreliable examples $\{z_i : \mathrm{Rel}(z_i) < b\}$ whereas in the set S_2 there are reliable examples $\{z_i : \mathrm{Rel}(z_i) \geq b\}$. In an ideal case when both splits are pure, the result would be equal to the entropy of classifications $H(S)$.

All experiments were performed by leave-one-out testing. In this setup, one example was reserved, while learning and preparatory calculations were performed on the rest, in many cases two nested leave-one-out testings were carried out. Final results are averages of leave-one-out experiments on all examples from the dataset.

Finally, we also applied our approach to a real-world application on a large database of 600.000 customers of a large local corporation. Here, due to large quantities of data, the testing methodology was slightly different. While leave-one out testing was still used for obtaining strangeness values for the training (50%) dataset, the remaining data was used as an independet testing set.

3 Results

Experimental results were obtained with two different setups. The first one consists of series of experiments on well-known (UCI) problem domains. These results were used to validate our approach and compare it with existing ones. The second experimental setup consists of applications in a real-life commercial data mining system. It also presents some valuable practical considerations.

Table 1 Comparison of confidence estimation on KNN with the algorithm-specific TCM-NN, both with 10 nearest neighbours. Accurracy was obtained with a standard 10-NN algorithm.

	Accuracy	Correlation with correctness		Information gain	
	KNN (%)	TCM-NN	KNN	TCM-NN	KNN
w Mesh	64.7	0.49	0.40	0.26	0.19
Brest cancer	80.2	0.09	0.14	0.02	0.03
Nuclear	81.0	0.35	0.28	0.12	0.07
Diabetes	73.7	0.26	0.19	0.06	0.05
Heart	79.3	0.34	0.18	0.11	0.09
Hepatitis	85.2	0.28	0.25	0.07	0.07
Iris	94.7	0.23	0.36	0.12	0.12
Chess end.	92.0	0.43	0.33	0.21	0.12
LED	73.2	0.20	0.19	0.04	0.05
Lymphography	83.1	0.50	0.22	0.32	0.18
Primary turnour	41.3	0.10	0.37	0.00	0.19
Rheumatology	61.3	0.42	0.42	0.17	0.16
Soybean	92.1	0.32	0.38	0.12	0.12
Voting	94.0	0.42	0.26	0.18	0.09
Average	78.3	0.32	0.28	0.13	0.11

3.1 Experiments on benchmark problems

Results of confidence estimation on the KNN (nearest neighbour) algorithm are compared with the TCM-NN nearest neighbour confidence machine [19], where a tailor-made strangeness measure for confidence estimation in a typicalness framework was constructed. In Table 1 experimental results in 15 domains are shown. Results of TCM-NN are slightly better, as could be expected from the tailor-made method, though the differences are not significant with two-tailed, paired t-test).

3.1.1 Reliability, confidence and density estimation

The obtained confidence values are compared with transductive reliability estimations and density estimations. Our first goal was to evaluate the performance of confidence values in terms of correlation with correctness, and its ability to separate correct and incorrect classifications in terms of information gain. Our second goal was to see whether confidence values are more easily interpretable than transductive reliability estimations.

Figures 3a and b depict how reliability estimations are transformed to confidence levels. This is a typical example and probably the most important result of our work, as it makes them easily statistically interpretable. On average, the best decision boundary for reliability estimations is 0.74, on the other hand, for confidence it is about 0.45. Also, the mass of correct and incorrect classification has shifted towards 1 and 0, respectively.

In Table 2 experimental results are presented. We see that confidence values significantly ($p < 0.05$ with two-tailed, paired t-test) outperform reliability estimations in terms of correlation with correctness. From Fig. 3 it is clear that this is because of the shift towards 1 and 0. Information gains do not differ significantly.

Table 2 Experimental results with confidence values, reliability and density estimations with 6 machine learning algorithm in 15 datasets. Accuracy was calculated as an average of all 6 base classifiers.

		Correlation with correctness			Information gain (in bit)		
Domain	Accuracy (%)	Reliability	Confidence	Density	Reliability	Confidence	Density
Mesh	65.7	0.51	0.46	0.10	0.25	0.25	0.12
Brest cancer	77.4	0.28	0.22	0.09	0.10	0.10	0.07
Nuclear	88.0	0.21	0.21	0.09	0.07	0.08	0.07
Diabetes	74.3	0.26	0.33	0.08	0.18	0.18	0.05
Heart	80.7	0.26	0.27	0.08	0.11	0.11	0.11
Hepatitis	86.6	0.25	0.30	0.12	0.12	0.11	0.14
Iris	93.8	0.23	0.42	0.08	0.13	0.13	0.09
Chess endgame	95.5	0.09	0.27	0.08	0.11	0.11	0.05
Chess endgame	71.1	0.11	0.12	0.07	0.10	0.10	0.05
LED	73.0	0.16	0.18	0.04	0.05	0.05	0.03
Lymphography	81.9	0.20	0.27	0.13	0.13	0.13	0.17
Primary tumor	44.8	0.39	0.38	0.07	0.16	0.16	0.07
Rheumatology	58.0	0.47	0.48	0.10	0.22	0.22	0.10
Soybean	89.4	0.35	0.37	0.08	0.14	0.13	0.09
Voting	94.0	0.17	0.22	0.09	0.08	0.08	0.07
Average	78.3	0.26	0.30	0.09	0.13	0.13	0.08

Comparing confidence values and density estimations shows a slightly different picture. Here, in terms of correlation with correctness as well as for for information gain criterion, the differences are significant ($p < 0.01$ with two-tailed, paired t-test). Figure 2 depicts typical density estimations for both correct

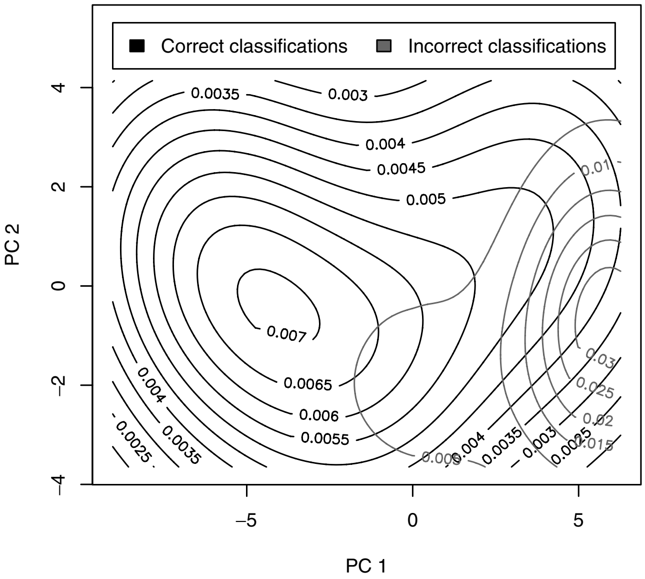
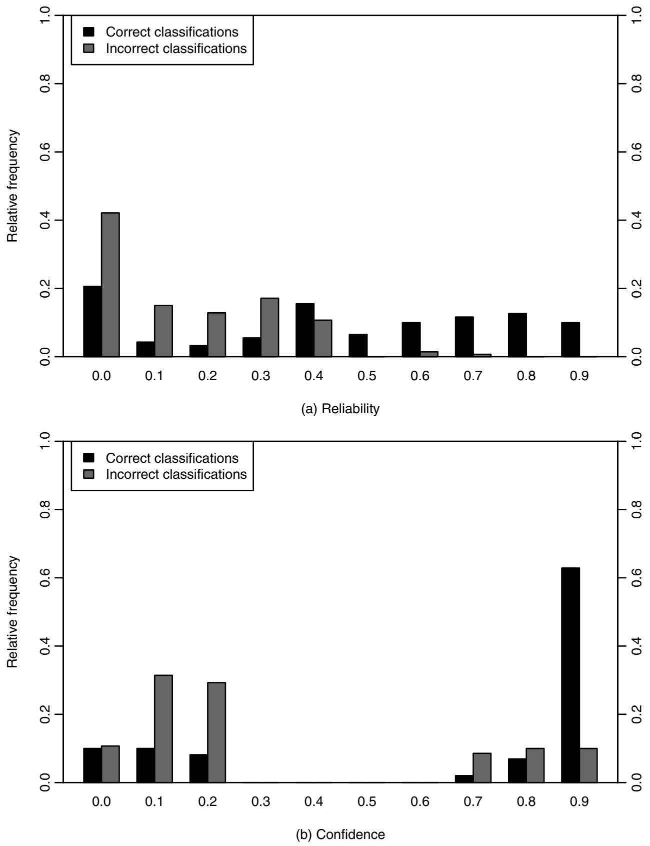


Fig. 2 Densities of correct and incorrect classification in soybean dataset using neural networks



(a) Reliability

(b) Confidence

Fig. 3 Relative frequencies of reliability estimations and confidence levels in soybean dataset using neural networks

and incorrect classifications. On average, the best decision boundary for density estimations is 0.52.

In Table 2 we can also see that meta-learning with 10 nearest neighbours (10-NN) performed worst (although 10 was a tuned parameter). This was expected, since it was used in the whole – sparsely populated – attribute space. Density estimations (Den.) performed significantly better on a reduced attribute space ($p < 0.01$ with two-tailed, paired t-test). We also see that transductive attributes improve the performance of density estimation (Tr. den.) quite significantly ($p < 0.05$ with two-tailed, paired t-test). While it does not reach performance of transductive reliability or confidence estimations, it is much easier to compute as it does not require re-learning of a classifier.

3.2 Real-life application and practical considerations

We also did a practical application of integration of decision support system with data mining methods working with data from extensive customer relationship management (CRM) survey for a large local corporation. It turned out that immense quantities of raw data had been collected and needed to be assessed. Thus the use of data mining methods was called for. The system was implemented in Oracle 9i application framework using Oracle's Data Mining (ODM) database extension. An Adaptive Bayesian Network classifier was used. The database consisted of about 600,000 customers' records consisting of up to 100 attributes. The preparatory calculations (leave-one-out testing on training dataset) were quite lengthy as they took more than a week. However, producing a confidence estimation for a single customer was much more acceptable; depending on system use it took about a minute.

Produced confidence values were much better (on average by 0.2 bit of gained information) than the probability estimations of the applied Adaptive Bayesian Network classifier. There was also improvement of more than 10% of the confident classification (confidence $\geq 95\%$). In practice this could (and in near future probably will) save significant amounts of CRM campaign money.

The main drawback of our approach in this practical problem is its relative slowness. It needs more than a week to perform preparatory calculations and it took again more than a week to calculate confidence values for all testing examples (customer records from independent set). It may therefore not be suitable for quick on-line analysis of an overall situation, but is perfectly suited for assessment of individual customers. A great advantage of typicalness/transduction approach over other approaches (such as kernel density estimation) is that it can be easily implemented even with relatively closed (no source code available for modifications) commercial data mining systems.

4 Discussion

We propose an approach that compensates the weaknesses of typicalness-based confidence estimation and transductive reliability estimation by integrating them into a joint confidence machine.

The resulting values are true confidence levels, and this makes them much easier to interpret. Contrary to the basic typicalness and transductive confidence estimation, the described approach is not bound to the particular underlying classifier. This is an important improvement since this makes possible to calculate confidence values for almost any classifier, no matter how complex it is.

Experimental comparison on comparable unmodified and modified algorithms (confidence estimation on a KNN algorithm and TCM-NN nearest neighbour confidence machine) show that the proposed approach performs similarly to the specially modified algorithm. There is no significant reduction in performance while there is a huge gain in generality.

Comparisons with kernel density estimation show that the computed confidence values significantly outperform density estimations. However, this does not mean that density estimations should not be used as they are much easier to compute and do not require re-learning of a classifier. Their performance can also be significantly improved by using additional transductive attributes.

Experimental results performed with different machine learning algorithms in several problem domains show that there is no reduction of discrimination performance with respect to transductive reliability estimation. More important than this, statistical interpretability of confidence values makes it possible to use applications in risk-sensitive problems with strict confidence limits.

The main drawback of our approach is computational complexity, as it needs to perform the leave-one-out testing in advance, and requires temporary re-learning of a classifier for each new example. However, this may not be a problem if incremental learners (such as naive Bayesian classifier) are used. In other cases, density estimation with included transductive attributes may also be used.

In the near future we are planning several experiments in risk-sensitive business problems as well as in medical diagnostics and prognostics.

Acknowledgements I thank anonymous reviewers for their very useful comments and suggestions. This work was supported by the Slovenian Ministry of Education and Science.

References

1. Bay SD, Pazzani MJ (2000) Characterizing model errors and differences. In: Proc. 17th international conf. on machine learning. Morgan Kaufmann, San Francisco, CA, pp 49–56
2. Blum A, Mitchell T (1998) Combining labeled and unlabeled data with co-training. In: Bartlett P, Mansour Y (eds) Proceedings of the 11th annual conference on computational learning theory. ACM Press, New York, USA, Madison, Wisconsin, pp 92–100
3. Diamond GA, Forester JS (1979) Analysis of probability as an aid in the clinical diagnosis of coronary artery disease. New England Journal of Medicine 300: 1350
4. Gammerman A, Vovk V, Vapnik V (1998) Learning by transduction. In: Cooper GF, Moral S (eds) Proceedings of the 14th conference on uncertainty in artificial intelligence. Morgan Kaufmann, San Francisco, USA, Madison, Wisconsin, pp 148–155
5. Gibbs AL, Su FE (2002) On choosing and bounding probability metrics. International Statistical Review 70(3): 419–435
6. Halck OM (2002) Using hard classifiers to estimate conditional class probabilities. In: Elomaa T, Mannila H, Toivonen H (eds) Proceedings of the thirteenth European conference on machine learning. Springer-Verlag, Berlin, pp 124–134
7. Hastie T, Tibisharani R, Friedman J (2001) The elements of statistical learning. Springer-Verlag
8. Ho SS, Wechsler H (2003) Transductive confidence machine for active learning. In: Proc. Int. joint Conf. on neural networks'03. Portland, OR
9. John GH, Langley P (1995) Estimating continuous distributions in Bayesian classifiers. In: Besnard P, Hanks S (eds) Proceedings of the eleventh conference on uncertainty in artificial intelligence. Morgan Kaufmann, San Francisco, USA
10. Kononenko I (1991) Semi-naive Bayesian classifier. In: Kodratoff Y (ed) Proc. European working session on learning-91. Springer-Verlag, Berlin-Heidelberg-New York, Porto, Potrugal, pp 206–219
11. Kukar M (2001a) Estimating classifications' reliability. PhD thesis, University of Ljubljana, faculty of computer and information science, Ljubljana, Slovenia. In Slovene
12. Kukar M (2001b) Making reliable diagnoses with machine learning: A case study. In: Quaglini S, Barahona P, Andreassen S (eds) Proceedings of artificial intelligence in medicine Europe, AIME 2001. Springer-Verlag, Berlin, Cascais, Portugal, pp 88–96
13. Kukar M, Kononenko I (2002) Reliable classifications with machine learning. In: Elomaa T, Mannila H, Toivonen H (eds) Proceedings of 13th European conference on machine learning. ECML 2002', Springer-Verlag, Berlin, pp 219–231
14. Li M, Vitányi P (1997) An introduction to Kolmogorov complexity and its applications. 2nd edn. Springer-Verlag, New York
15. Melluish T, Saunders C, Nouretdinov I, Vovk V (2001) Comparing the Bayes and typicalness frameworks. In: Proc. ECML 2001. vol 2167, pp 350–357

16. Nouretdinov I, Melluish T, Vovk V (2001) Ridge regressioon confidence machine. In: Proc. 18th international conf. on machine learning. Morgan Kaufmann, San Francisco, CA, pp 385–392
17. Olona-Cabases M (1994) The probability of a correct diagnosis. In: Candell-Riera J, Ortega-Alcalde D (eds) Nuclear cardiology in everyday practice. Kluwer, Dordrecht, NL, pp 348–357
18. Pfahringer B, Bensuasan H, Giraud-Carrier C (2000) Meta-learning by landmarking various learning algorithms. In: Proc. 17th international conf. on machine learning. Morgan Kaufmann, San Francisco, CA
19. Proedrou K, Nouretdinov I, Vovk V, Gammerman A (2002) Transductive confidence machines for pattern recognition. In: Proc. ECML 2002. Springer, Berlin, pp 381–390
20. Robnik-Šikonja M, Kononenko I (2003) Theoretical and empirical analysis of ReliefF and RReliefF. Mach. learn. 53: 23–69
21. Rumelhart D, McClelland JL (1986) Parallel distributed processing, vol 1: Foundations. MIT Press, Cambridge
22. Saunders C, Gammerman A, Vovk V (1999) Transduction with confidence and credibility. In: Dean T (ed) Proceedings of the international joint conference on artificial intelligence. Morgan Kaufmann, San Francisco, USA, Stockholm, Sweden
23. Seewald A, Furnkranz J (2001) An evaluation of grading classifiers. In: Proc. 4th international symposium on advances in intelligent data analysis. pp 115–124
24. Smyth P, Gray A, Fayyad U (1995) Retrofitting decision tree classifiers using kernel density estimation. In: Prieditis A, Russell SJ (eds) Proceedings of the twelvth international conference on machine learning. Morgan Kaufmann, San Francisco, USA, Tahoe City, California, USA, pp 506–514
25. Specht DF, Romsdahl H (1994) Experience with adaptive pobabilistic neural networks and adaptive general regression neural networks. In: Rogers SK (ed) Proceedings of IEEE international conference on neural networks. IEEE Press, Piscataway, USA, Orlando, USA
26. Taneja IJ (1995) On generalized information measures and their applications. Adv. Electron. Elect. Physics 76: 327–416
27. Vapnik V (1998) Statistical learning theory. John Wiley, New York, USA
28. Venables WN, Ripley BD (2002) Modern applied statistics with S-PLUS, 4th ed. Springer-Verlag
29. Vovk V, Gammerman A, Saunders C (1999) Machine learning application of algorithmic randomness. In: Bratko I, Dzeroski S (eds) Proceedings of the 16th international conference on machine learning (ICML'99). Morgan Kaufmann, San Francisco, USA, Bled, Slovenija
30. Wand MP, Jones MC (1995) Kernel smoothing. Chapman and Hall, London

Matjaž Kukar is currently Assistant Professor in the Faculty of Computer and Information Science at University of Ljubljana. His research interests include machine learning, data mining and intelligent data analysis, ROC analysis, cost-sensitive learning, reliability estimation, and latent structure analysis, as well as applications of data mining in medical and business problems.