Kull, M., & Flach, P. A. (2014). Reliability Maps: A Tool to Enhance Probability Estimates and Improve Classification Accuracy (Best paper award). In T. Calders, F. Esposito, E. Hullermeier, & R. Meo (Eds.), Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part II. (pp. 18-33). (Lecture Notes in Artificial Intelligence; Vol. 8725). Springer Berlin Heidelberg, 2010. DOI: 10.1007/978-3-662-44851-9_2

# Reliability Maps: A Tool to Enhance Probability Estimates and Improve Classification Accuracy

Meelis Kull and Peter Flach

Intelligent Systems Laboratory, University of Bristol, United Kingdom
{Meelis.Kull,Peter.Flach}@bristol.ac.uk

**Abstract.** We propose a general method to assess the reliability of two-class probabilities in an instance-wise manner. This is relevant, for instance, for obtaining calibrated multi-class probabilities from two-class probability scores. The LS-ECOC method approaches this by performing least-squares fitting over a suitable error-correcting output code matrix, where the optimisation resolves potential conflicts in the input probabilities. While this gives all input probabilities equal weight, we would like to spend less effort fitting unreliable probability estimates. We introduce the concept of a reliability map to accompany the more conventional notion of calibration map; and LS-ECOC-R which modifies LS-ECOC to take reliability into account. We demonstrate on synthetic data that this gets us closer to the Bayes-optimal classifier, even if the base classifiers are linear and hence have high bias. Results on UCI data sets demonstrate that multi-class accuracy also improves.

## 1 Introduction

Classification problems can be approached using a range of machine learning models. Some of these models, including decision trees, naive Bayes and nearest neighbour, deal naturally with more than two classes. Others – most notably linear models and their kernelised variants – are essentially two-class or binary. In order to solve a multi-class problem with binary models we need to decompose the multi-class problem into a set of binary subproblems, train a classifier on each subproblem and aggregate the predicted classes or scores obtained on each subproblem into an overall multi-class prediction or score vector. In the most common scenarios these subproblems are either pairwise (one class against another class) or one-vs-rest (one class against all other classes), which in matrix form could be described as follows:

$$\mathbf{M} = \begin{pmatrix} +1 & +1 & 0 \\ -1 & 0 & +1 \\ 0 & -1 & -1 \end{pmatrix} \qquad \mathbf{N} = \begin{pmatrix} +1 & -1 & -1 \\ -1 & +1 & -1 \\ -1 & -1 & +1 \end{pmatrix}$$

These are known as code matrices, with binary subproblems in columns and classes in rows. $\mathbf{M}$ encodes pairwise subproblems and $\mathbf{N}$ encodes one-vs-rest. A vector of outputs from the binary classifiers can often be traced back to one of the classes: e.g., if we receive $(+1,+1,-1)$ in the pairwise case we can construe this as two votes for the first class and one vote for the third class.

The general approach of error-correcting output codes was pioneered by [4] and later refined to take classifier scores into account [8, 1]. Error-correcting capability is achieved by building redundancy into the code matrix: for instance, we can use two different binary classifiers for each subproblem, leading to two copies of each column in the code matrix (in fact, ensemble methods can be represented by a code matrix with repeated columns). More generally, ECOC can be described as an approach to combining the opinions of many different experts. Each expert has its own group of positive and negative classes (not necessarily covering all classes) and is trained to decide whether an unlabelled example falls in the positive or negative group. Expert opinions may disagree, in which case we need to figure out how to tweak the opinions to agree, and possibly which experts to trust more than others. Therefore, it is important to know how reliable or confident each expert is. For example, a properly Bayesian expert would output a posterior probability distribution over all possible opinions, from which we can infer a confidence level (e.g., expressed as a variance). However, while most machine learning models can be made to output a (more or less calibrated) probability score, they rarely give information about their confidence, and so a model-independent method needs to learn the reliability of these scores. Note that a calibrated probability score quantifies the expert's uncertainty in the class value, but here we are after the uncertainty in that probability estimate. That is, a weather forecaster can be very certain that the chance of rain is 50%; or her best estimate at 20% might be very uncertain due to lack of data.

This paper proposes a practical method to learn the reliability of probability scores output by experts in the above scenario, in an instance-wise manner. Being able to assess the reliability of a probability score for each instance is much more powerful than assigning an aggregate reliability score to each expert, independent of the instance to be classified. For example, we show later that an ECOC-based method that takes instance-wise reliability into account allows us to learn non-linear decision boundaries even when employing linear base models. As such the method can be seen as reducing the bias of the base classifier. But the basic method has applicability beyond ECOC. For example, in comparison with another bias-reducing technique, boosting [12], which uses a single confidence factor per base classifier, our method offers the possibility to generalise this to instance-wise confidence which should result in a better model. The advantage of having calibrated probability estimates in a cost-based scenario is that we can better minimise expected overall cost by predicting the class that minimises the cost averaged over all possible true classes. Taking reliability of the probability scores into account gives us the choice of choosing a non-minimising class if it has the benefit of less uncertainty. This would be useful in the presence of hard constraints of the form 'the probability that the cost exceeds budget $B$ must be less than 5%' which may be true for a class even if it does not minimise expected cost.

The outline of the paper is as follows. Section 2 introduces reliability maps and their relation to squared bias of probability estimates from the respective true posterior probabilities. Section 3 develops an algorithm to learn reliability maps from class-labelled data, without access to true posterior probabilities. Section 4 introduces LS-ECOC-R, a reliability-weighted version of the LS-ECOC method to obtain multi-class probability scores. In Section 5 we present two kinds of experiments: we investigate how far our

estimates are from the truth on synthetic data, and we investigate the effect of using reliabilities on the quality of multi-class predictions and probability scores. Section 6 discusses related work, and Section 7 concludes.

## 2 Calibration and Reliability

Let $X, Y$ be the random variables representing the unknown true model of our binary classification task. That is, $X$ is a random variable over the instance space $\mathscr{X}$ and $Y$ is a binary random variable with 1 and 0 standing for positive and negative classes, respectively. Ideally, we would like to know the true positive class posterior $q(x)$ for each possible $x \in \mathscr{X}$:

$$q(x) = P(Y{=}1|X{=}x). \tag{1}$$

In reality, we use training data to learn a model $g : \mathscr{X} \to [0,1]$ such that $g(x)$ is approximating $q(x)$. If the output of the model is $g(x) = s$, what can we say about the true value $q(x)$? Let $\mu_g(s)$ be the expected proportion of positives among all instances $x$ with the same value $g(x) = s$:

$$\mu_g(s) = \mathbb{E}[q(X)|g(X){=}s] = P(Y{=}1|g(X){=}s). \tag{2}$$

The function $\mu_g$ is known as the (true) *calibration map* of the probability estimator $g$. If the estimator $g$ is perfectly calibrated, then $\mu_g$ is the identity function. If not, then there are many methods of calibration which can be applied to learn an estimated calibration map $\hat{\mu}_g$ such that $\hat{\mu}_g(g(x))$ is approximately equal to $\mu_g(g(x))$. However, for individual instances $x$ the expected proportion of positives $q(x)$ can deviate from the mean proportion $\mu_g(g(x))$, i.e. the following variance is non-zero:

$$\sigma_g^2(s) = \mathsf{var}[q(X)|g(X){=}s] = \mathbb{E}[(q(X) - \mu_g(s))^2|g(X){=}s]. \tag{3}$$

The magnitude of $\sigma_g^2(s)$ across the estimates $s$ from the model $g$ actually determines how useful $g$ is for estimating $q$. For instance, a constant probability estimator $g(x) = P(Y{=}1)$ is perfectly calibrated, but has high $\sigma_g^2(s)$ for its constant estimate $s = P(Y{=}1)$. The perfect estimator $g(x) = q(x)$ has $\sigma_g^2(s) = 0$ for all $s$. The variance $\sigma_g^2$ is bounded from above by $\sigma_g^2(s) \leq \mu_g(s) \cdot (1 - \mu_g(s))$, where the equality holds when $q(x)$ is either 0 or 1 for each $x$ with $g(x) = s$.[1] This leads to our following definition of the *reliability map* $r_g$ of the probability estimator $g$.

**Definition 1.** *Let $g : \mathscr{X} \to [0,1]$ be an estimator of probability $q(x) = P(Y{=}1|X{=}x)$. Then the* reliability map *$r_g$ of the probability estimator $g$ is defined as follows:*

$$r_g(s) = 1 - \frac{\sigma_g^2(s)}{\mu_g(s) \cdot (1 - \mu_g(s))}, \tag{4}$$

[1] This can be seen as follows: $\sigma_g^2(s) = \mathbb{E}[(q(X))^2|g(X) = s] - (\mu_g(s))^2 \leq \mathbb{E}[q(X)|g(X) = s] - (\mu_g(s))^2 = \mu_g(s) - (\mu_g(s))^2 = \mu_g(s) \cdot (1 - \mu_g(s))$. The equality holds when $P[(q(X))^2 = q(X)|g(X) = s] = P[q(X) \in \{0,1\}|g(X) = s] = 1$.

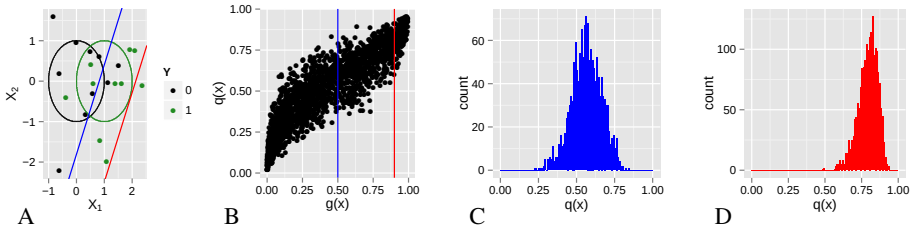


**Fig. 1.** Synthetic 2-class data of Gaussians centered at (0,0) and (1,0). (A) Standard deviation circles of the two Gaussians, 20 training instances and lines $g(x) = 0.5$ (blue) and $g(x) = 0.9$ (red) where $g$ is the learned logistic regression model; (B) scatterplot of $q(x)$ and $g(x)$ for 2000 test points; (C) histogram of $q(x)$ for the 1530 test points out of 200000 which have $g(x)$ in range $[0.495, 0.505]$; (D) histogram of $q(x)$ for the 1727 test points out of 200000 with $g(x)$ in $[0.895, 0.905]$.

*where the calibration map* $\mu_g(s)$ *and variance* $\sigma_g^2(s)$ *are defined by* (2) *and* (3)*. For an estimate* $s$ *from the model* $g$*, we refer to the value* $r_g(s)$ *as* local reliability *of* $g$ *at* $s$*.*

Minimum and maximum values 0 and 1 for the local reliability mean respectively that $q(x) \in \{0,1\}$ and $q(x) = g(x)$ for all $x$ with $g(x) = s$. We call it local reliability so that we can still talk about the (global) reliability of the probability estimator as a whole. Figure 3 presents the calibration and reliability maps for a synthetic dataset described in Section 5.

*Example 1.* To illustrate the above notions consider a synthetic two-class generative model with uniform class distribution $P(Y = 1) = P(Y = 0) = 1/2$ and $X$ distributed as a standard 2-dimensional normal distribution centered at $(1,0)$ for class $Y = 1$ and at $(0,0)$ for class $Y = 0$. We generated 20 training instances from this generative model and learned a logistic regression model $g : \mathbb{R}^2 \to [0,1]$ to estimate posterior class probabilities $q(x) = P(Y = 1|X = x)$, see Fig. 1A. In this experiment our logistic regression learner resulted in the model $g(x) = 1/(1+\exp(1.37 - 1.68x_1 + 0.76x_2))$ whereas the true model is $q(x) = P(Y{=}1|X{=}x) = 1/(1+\exp(0.5 - x_1))$. This implies that for any instance $x$ the learned estimate $g(x)$ can deviate slightly from the true value $q(x)$, see Fig. 1B with 2000 test points drawn randomly from the same generative model with two Gaussians.

Consider now the group of all instances with $g(x) = 0.5$, located on the blue line in Fig. 1AB. The histogram of $q(x)$ for a sample of these points is given in Fig. 1C with mean $\mu_g(0.5) = 0.5675$ and estimated variance $\sigma_g^2(0.5) = 0.0101$, leading to a reliability value of $r_g(s) = 0.9589$. What this demonstrates is that at predicted score 0.5 there is little variation in the true probabilities, even though the estimator is not perfectly calibrated at that score. For the group $g(x) = 0.9$ shown in red in Fig. 1AB and with a histogram of $q(x)$ in Fig. 1D the mean and variance are $\mu_g(x) = 0.7979$ and $\sigma_g^2(x) = 0.0042$, resulting again in a high reliability of $r_g(s) = 0.9740$.

The estimated $g(x) = s$ can differ from the true $q(x)$ for one or both of the following two reasons. First, if $\mu_g(s) \neq s$ then there is a bias in $g(x)$ from the average $q(x)$ of the group of instances with the same estimate $s$. Second, if $\sigma_g^2(s) > 0$ then there is variance in $q(x)$ for the group of instances with the same estimate $s$. In fact, the instance-wise squared loss between $g$ and $q$ within the group of instances with the same estimate $s$ can be decomposed into these two losses:

$$\begin{aligned}
&\mathbb{E}[(g(X)-q(X))^2|g(X)=s] = \\
&= \mathbb{E}[(s-q(X))^2|g(X)=s] = \mathbb{E}[(s-\mu_g(s)+\mu_g(s)-q(X))^2|g(X)=s] = \\
&= (s-\mu_g(s))^2 + 2(s-\mu_g(s))\mathbb{E}[\mu_g(s)-q(X)|g(X)=s] + \mathbb{E}[(\mu_g(s)-q(X))^2|g(X)=s] = \\
&= (s-\mu_g(s))^2 + \sigma_g^2(s),
\end{aligned}$$

where the last equality holds because $\mathbb{E}[\mu_g(s) - q(X)|g(X)=s] = 0$. This decomposition can be averaged over the whole instance space, resulting in the following decomposition:

$$\mathbb{E}[(g(X)-q(X))^2] = \mathbb{E}[(g(X)-\mu_g(g(X)))^2] + \mathbb{E}[(\mu_g(g(X))-q(X))^2].$$

We will refer to these three quantities as *instance-wise calibration loss*, *group-wise calibration loss*, and *grouping loss*.[2] Any calibration procedure which transforms values of $g(x)$ with a calibration map can decrease the group-wise calibration loss but not the grouping loss, which is inherent to the model. Grouping loss arises from the model's decision to group certain instances together with the same probability estimate whereas the true probabilities are different. The quantity $\sigma_g^2(s)$ can be interpreted as the local grouping loss for one group of instances with the same estimate $g(x) = s$ and the total grouping loss is the average $\sigma_g^2(s)$ across all groups $s$:

$$\mathbb{E}[(\mu_g(g(X)) - q(X))^2] = \mathbb{E}[\sigma_g^2(g(X))].$$

*Example 1 (continued).* In the example of Fig. 1, the group $g(x) = 0.5$ suffers instance-wise calibration loss equal to 0.0147 decomposing into group-wise calibration loss of $(0.5675-0.5)^2 = 0.0046$ and grouping loss of $\sigma_g^2(s) = 0.0101$. Calibration of $g$ can decrease the group-wise calibration loss, but the grouping loss remains irreducible, unless a new model is trained instead of $g$.

## 3 Learning Calibration and Reliability Maps

Learning calibration maps is a task that has been solved earlier with various methods. One simple approach that we revisit below is to view the binary label $Y$ as a dependent variable and the probability estimate $S = g(X)$ as the independent variable and apply any standard regression learning algorithm. The training data for such approach is a list of pairs $(S_i, Y_i)$. Although each individual instance $S_i$ is far from the true calibration

[2] The instance-wise calibration loss bears similarity to the calibration loss which is obtained by decomposing the Brier score [10], but the difference is that there the comparison is made with the empirical probability rather than the true probability.

map, the expected value of $Y$ given a fixed estimate $S$ lies at the calibration map (see (6) below). In other words, $Y$ is an unbiased estimator of the calibrated probability. Assuming that the true calibration map is continuous, this allows to estimate it with regression.

For learning reliability maps it is also possible to use regression, but the challenge is to come up with a suitable unbiased estimator. Since the Bernoulli distribution of the binary label $Y$ given an estimate $S$ has only one parameter determining the calibrated probability, it does not contain information about the reliability map. For each instance $X$ we need more information about the true probability $q(X)$ than just a single binary label $Y$. Our solution is to gather a small group of similar instances $X_1,\dots,X_m$ with approximately the same estimate $g(X_i) \approx S$ and approximately the same posterior $q(X_i) \approx Q$. We obtain such groups of instances by splitting the training instances into clusters of equal size $m$ according to some distance measure in the instance space. The clustering method that we have used in the experiments to obtain clusters of size $m = 10$ is described in Section 3.1 below. The reason for building clusters is that the variance in the number of positives $\sum_{i=1}^{m} Y_i$ in a cluster contains information about the variance in the posterior, $\sigma_g^2(S)$. As an estimator of local reliability of $g$ at $S$ we use $R^{(m)}$, defined as follows:

$$R^{(m)} = 1 + \frac{1}{m-1} - \frac{(\sum_{i=1}^{m} Y_i - m\mu_g(S))^2}{m(m-1)\mu_g(S)(1-\mu_g(S))}. \tag{5}$$

Theorem 1 proves by equality (7) that this estimator is unbiased if the instances within the cluster have equal $g$ and equal $q$.

**Theorem 1.** *Let $g : \mathscr{X} \to [0,1]$ be a fixed probability estimator and let $(X_i, Y_i)$ for $i = 1,\dots,m$ with $m \geq 2$ be an i.i.d. random sample distributed identically to $(X,Y)$ where $X$ is a random variable over $\mathscr{X}$ and $Y$ is a binary random variable. Additionally, let $\mathscr{C}$ stand for the condition where $g(X_i) = g(X)$ and $q(X_i) = q(X)$ for $i = 1,\dots,m$, where $q$ is defined as in (1). Then the following two equalities hold:*

$$\mu_g(s) = \mathbb{E}[Y|S{=}s] \tag{6}$$

$$r_g(s) = \mathbb{E}[R^{(m)}|S{=}s,\mathscr{C}] \tag{7}$$

*where $S = g(X)$ and $\mu_g$, $r_g$, $R^{(m)}$ are defined above respectively in (2), (4) and (5).*

*Proof.* Equation (6) can easily be proved by denoting $Q = q(X)$ and applying the law of total expectation:

$$\mathbb{E}[Y|S{=}s] = \mathbb{E}[\mathbb{E}[Y|Q,S{=}s]|S{=}s] = \mathbb{E}[Q|S{=}s] = \mu_g(s).$$

Let us denote $Z = \sum_{i=1}^{m} Y_i$. As $Y_i$ are independent given $\mathscr{C}$ then $\mathbb{E}[Z|S{=}s,\mathscr{C}] = m\mu_g(s)$. Therefore,

$$\mathbb{E}[R^{(m)}|S{=}s,\mathscr{C}] = 1 + \frac{1}{m-1} - \frac{\mathsf{var}[Z|S{=}s,\mathscr{C}]}{m(m-1)\mu_g(s)(1-\mu_g(s))}.$$

Due to (4) it now remains to prove that

$$\frac{\mathsf{var}[Z|S{=}s,\mathscr{C}]}{m(m-1)\mu_g(s)(1-\mu_g(s))} = \frac{\sigma_g^2(s)}{\mu_g(s)(1-\mu_g(s))} + \frac{1}{m-1},$$

or equivalently, that

$$\mathsf{var}[Z|S{=}s,\mathscr{C}] = m(m-1)\sigma_g^2(s) + m\mu_g(s)(1-\mu_g(s)).$$

Let us denote $Q = q(X_i)$. As $Z$ is binomially distributed given $Q$, $S{=}s$ and $\mathscr{C}$, we have $\mathbb{E}[Z|Q,S{=}s,\mathscr{C}] = mQ$ and $\mathsf{var}[Z|Q,S{=}s,\mathscr{C}] = mQ(1-Q)$. Also, $\mathbb{E}[Q|S{=}s,\mathscr{C}] = \mu_g(s)$ and $\mathsf{var}[Q|S{=}s,\mathscr{C}] = \sigma_g^2(s)$ and $\mathbb{E}[Q^2|S{=}s,\mathscr{C}] = \mathsf{var}[Q|S{=}s,\mathscr{C}] + (\mathbb{E}[Q|S{=}s,\mathscr{C}])^2 = \sigma_g^2(s) + \mu_g^2(s)$. Using this and the law of total variance (and algebraic manipulations) we obtain the following:

$$\begin{aligned}
\mathsf{var}[Z|S{=}s,\mathscr{C}] &= \mathbb{E}[\mathsf{var}[Z|Q,S{=}s,\mathscr{C}]] + \mathsf{var}[\mathbb{E}[Z|Q,S{=}s,\mathscr{C}]] = \\
&= \mathbb{E}[mQ(1-Q)|S{=}s,\mathscr{C}] + \mathsf{var}[mQ|S{=}s,\mathscr{C}] = \\
&= m\mathbb{E}[Q|S{=}s,\mathscr{C}] - m\mathbb{E}[Q^2|S{=}s,\mathscr{C}] + m^2\mathsf{var}[Q|S{=}s,\mathscr{C}] = \\
&= m\mu_g(s) - m\sigma_g^2(s) - m\mu_g^2(s) + m^2\sigma_g^2(s) = \\
&= m\mu_g(s)(1-\mu_g(s)) + m(m-1)\sigma_g^2(s),
\end{aligned}$$

which completes the proof. □

In practice, the estimates $g$ and true probabilities $q$ are equal for a cluster only approximately, so the equality (7) also holds only approximately. Due to clustering the number of training instances for regression is $m$ times smaller than for the original problem, so learning the reliability map is harder than learning the calibration map. However, the experiments show that with a training set of 2000 instances the learned reliability map can be already accurate enough to improve multi-class probability estimation and classification. Next we describe what regression and clustering methods we are using to achieve this.

### 3.1 Regression and Clustering Methods for Learning the Maps

First let us stress that there is a wide variety of regression and clustering methods and many could be used for learning calibration and reliability maps. The choice has certainly implications on the performance of multi-class probability estimation and classification, but the comparison of different methods remains as future work. Here we describe the methods we have chosen.

For regression we use local linear regression with the Epanechnikov kernel and fixed bandwidth. For learning the calibration map we have the training pairs $(S_i, Y_i)$ for $i = 1,\ldots n$. The regression estimate $\hat{\mu}_g(s)$ for a target point $s$ is calculated as follows:

$$\begin{aligned}
\hat{\mu}_g(s) &= \alpha(s) + \beta(s)\cdot s, \\
\alpha(s),\beta(s) &= \operatorname*{argmin}_{\alpha,\beta\in\mathbb{R}} \sum_{i=1}^{n} K_\lambda(s,S_i)\cdot(Y_i - \alpha - \beta\cdot S_i)^2
\end{aligned}$$

where $\lambda > 0$ is the fixed bandwidth of the Epanechnikov kernel $K_\lambda$ defined as follows:

$$K_\lambda(s, S_i) = \begin{cases} \frac{3}{4}(1-(s-S_i)^2) & \text{if } |s-S_i| \leq 1; \\ 0 & \text{otherwise.} \end{cases}$$

For learning the reliability map we use the same method, that is:

$$\hat{r}_g(s) = \alpha'(s) + \beta'(s) \cdot s,$$
$$\alpha'(s), \beta'(s) = \underset{\alpha', \beta' \in \mathbb{R}}{\operatorname{argmin}} \sum_{i=1}^{n} K_{\lambda'}(s, S_i) \cdot (R_i^{(m)} - \alpha' - \beta' \cdot S_i)^2.$$

Local linear regression can produce estimates outside of our range $[0,1]$ and this problem needs to be addressed. Actually, the extreme values 0 and 1 are also undesired, because they present an over-confident statement about the probabilities. In the experiments we used 0.001 and 0.999 as the lower and upper bound for regression and all estimates outside of this range were changed to these values.

For the clustering method we set the following requirements:

(a) the resulting clusters must all be of fixed size $m$;
(b) within each cluster the estimate $g$ should be approximately equal;
(c) within each cluster the true posterior $q$ should be approximately equal.

Our first step is to order the instances by $g$ and to cut the ordered list into *super-clusters* of size $k \cdot m$, in the experiments we used $k = 20$ and $m = 10$. With $n \gg k \cdot m$ every super-cluster satisfies requirement (b) to some extent. We then cluster each super-cluster into $k$ clusters of size $m$ according to some distance measure between the instances, in the experiments we used the Euclidean distance. Depending on how smooth $q$ is and how tightly together the instances are, the resulting clusters can satisfy the requirement (c) to some extent. A few instances can remain unclustered to satisfy the requirement (a).

To cluster $k \cdot m$ instances of a super-cluster into $k$ clusters of size $m$ according to some distance measure we modify the DIANA clustering algorithm for this purpose [7]. DIANA is a divisive algorithm which splits at each step one of the existing clusters into two. The splitting is initialised by creating an empty new cluster besides the existing one. Then the algorithm iterates and in each iteration reassigns one instance from the old cluster to the new one. For reassignment it chooses the instance with the largest value for the sum of distances to the instances of the old cluster minus to the new cluster. The original version of the algorithm stops reassignments when the respective value becomes negative, we stop when the size of the new cluster is divisible by $m$ and differs from the size of the old cluster by at most $m$. The original DIANA has to decide which cluster to split next, for us the order does not matter because of the required fixed size $m$. Our algorithm ends when all clusters are of size $m$, except one can be smaller. The smaller cluster is discarded from learning the reliability map.

## 4 LS-ECOC-R: Multi-Class Probability Estimation with Reliabilities

Next we show that the learned calibration and reliability maps $\hat{\mu}_g$ and $\hat{r}_g$ can be used for multi-class probability estimation with ECOC. The ECOC decomposition of a $K$-class

task into $L$ binary tasks is represented as a code matrix $M \in \{-1,0,+1\}^{K\times L}$. The binary task represented by column $l$ aims at discriminating between the positive group of classes $\mathscr{C}_l^+ = \{k|M_{k,l} = +1\}$ and the negative group of classes $\mathscr{C}_l^- = \{k|M_{k,l} = -1\}$. The neutral group of classes $\mathscr{C}_l^0 = \{k|M_{k,l} = 0\}$ is excluded from the training set for the $l$-th binary model. Suppose that for a given coding matrix $M$ we have trained $L$ binary probability estimators $g_l : \mathscr{X} \to [0,1]$ for tasks $l = 1,\ldots,L$, and learned their calibration maps $\hat{\mu}_{g_l}$ and reliability maps $\hat{r}_{g_l}$. LS-ECOC [8] estimates multi-class posterior probabilities by combining the calibrated probability estimates $\hat{\mu}_{g_l}(g_l(x))$ only.

First denote by $q_l(x)$ the true posterior of the positive group given that instance $x$ is not neutral:

$$q_l(x) = P(Y \in \mathscr{C}_l^+ | Y \in \mathscr{C}_l^\pm, X = x) = \frac{\sum_{k\in\mathscr{C}_l^+} p_k}{\sum_{k\in\mathscr{C}_l^\pm} p_k}$$

where $\mathscr{C}_l^\pm = \mathscr{C}_l^+ \cup \mathscr{C}_l^-$ and $p_k = P(Y = k|X = x)$. Let $\varepsilon_l$ be the error of $\hat{\mu}_{g_l}(g_l(x))$ in estimating the true $q_l(x)$:

$$\varepsilon_l = q_l(x) - \hat{\mu}_{g_l}(g_l(x))$$

The idea of LS-ECOC is to estimate the posterior probabilities $p_k$ such that the total squared error $\sum_{l=1}^L \varepsilon_l^2$ is minimised:

$$\hat{p} = \underset{\substack{p_k\geq 0\\ \sum p_k = 1}}{\operatorname{argmin}} \sum_{l=1}^L \varepsilon_l^2 = \underset{\substack{p_k\geq 0\\ \sum p_k = 1}}{\operatorname{argmin}} \sum_{l=1}^L \left( \frac{\sum_{k\in\mathscr{C}_l^+} p_k}{\sum_{k\in\mathscr{C}_l^\pm} p_k} - \hat{\mu}_{g_l}(g_l(x)) \right)^2$$

If $\sum_{k\in\mathscr{C}_l^\pm} p_k = 1$ for each $l$, that is if the coding matrix is actually binary, then this is a straightforward least-squares optimisation with linear constraints which is convex and can easily be solved to estimate $\hat{p}$. The optimisation for ternary coding matrices can in general be non-convex.

Effectively, LS-ECOC assumes that $\varepsilon_l$ is normally distributed around 0 with the same variance for all $l$. Therefore, LS-ECOC is equally confident in each value of $\hat{\mu}_{g_l}(g_l(x))$ regardless of which binary model it is resulting from and what the value of the estimate is. For example, the calibrated probability estimates 0.01 from model $g_1$ and 0.5 from model $g_2$ are equally likely to be off by 0.1 according to LS-ECOC.

This is where we can benefit from the learned reliability map $\hat{r}_{g_l}$. We propose a variant of LS-ECOC which we denote LS-ECOC-R (for LS-ECOC with reliability estimates). LS-ECOC-R assumes that $\varepsilon_l$ is normally distributed around 0 with variance $\hat{\sigma}_g^2(g_l(x))$ where $\hat{\sigma}_g^2$ is calculated due to (4) as follows:

$$\hat{\sigma}_g^2(s) = (1 - \hat{r}_g(s)) \cdot \hat{\mu}_g(s) \cdot (1 - \hat{\mu}_g(s)). \tag{8}$$

So there is potentially a different level of confidence in each probability estimate for each instance. The multi-class probability estimates with LS-ECOC-R are obtained as follows:

$$\hat{p} = \underset{\substack{p_k\geq 0\\ \sum p_k = 1}}{\operatorname{argmin}} \sum_{l=1}^L \frac{\varepsilon_l^2}{\hat{\sigma}_g^2(g_l(x))} = \underset{\substack{p_k\geq 0\\ \sum p_k = 1}}{\operatorname{argmin}} \sum_{l=1}^L \left( \frac{\sum_{k\in\mathscr{C}_l^+} p_k}{\hat{\sigma}_{g_l}(g_l(x)) \sum_{k\in\mathscr{C}_l^\pm} p_k} - \frac{\hat{\mu}_{g_l}(g_l(x))}{\hat{\sigma}_{g_l}(g_l(x))} \right)^2.$$

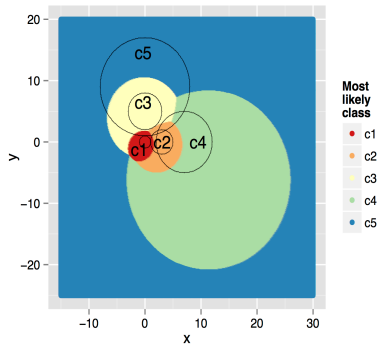


**Fig. 2.** Generative model for the synthetic dataset with 5 Gaussians with black circles centered at the means $\mu_c$ and radius equal to respective $\sigma_c$. Bayes-optimal decision regions are coloured by classes.

## 5 Experimental Evaluation

In the experiments we have three objectives. First, we demonstrate that the proposed learning methods can indeed provide good estimates of the calibration and reliability maps. For this we use a synthetic dataset where we know exactly the true calibration and reliability maps and can therefore compare our estimates to ground truth. Second, we show on the same dataset that LS-ECOC-R using the estimated reliability map outperforms LS-ECOC on probability estimation and classification. Finally, we show that LS-ECOC-R outperforms LS-ECOC also on 6 real datasets.

### 5.1 Experiments on Synthetic Data

As we need to know the true posterior distribution for an in-depth evaluation, we generate synthetic data with a probabilistic generative model. We use the same model that has earlier been used in several papers relating to multi-class probability estimation [16, 15, 18]. This generative model has 5 equiprobable classes and the instances of each class are distributed as a 2-dimensional normal distribution with the following parameters:

| | class 1 | class 2 | class 3 | class 4 | class 5 |
|---|---|---|---|---|---|
| $\mu_c$ | $(0,0)$ | $(3,0)$ | $(0,5)$ | $(7,0)$ | $(0,9)$ |
| $\sigma_c^2$ | 1 | 4 | 9 | 25 | 64 |

where $\mu_c$ is the mean and the covariance matrix is the unit matrix multiplied by $\sigma_c^2$. Figure 2 shows in colours the Bayes-optimal decision regions of this probabilistic model and the black circles are centered at the means of the Gaussians and have radius equal to the respective $\sigma_c$.

In order to evaluate our calibration and reliability learning algorithms we consider binary base estimators for which we can calculate the true calibration and reliability

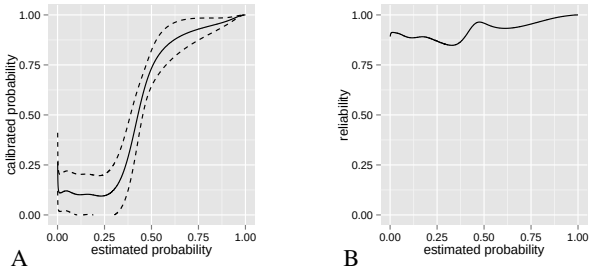


**Fig. 3.** Solid lines are the true (A) calibration map $\mu_{g_l}$ and (B) reliability map $r_{g_l}$ for the logistic regression model $g_l$ comparing classes 3 and 5 against others for the synthetic dataset. The dashed lines in (A) mark $\mu_{g_l} - \sigma_{g_l}$ and $\mu_{g_l} + \sigma_{g_l}$.

maps. This is possible whenever the contours of equal probability estimate are straight lines. Along any straight line the instances of each class have univariate Gaussian distribution, and we can analytically derive the class proportions and the parameters of the respective Gaussians. The true calibration maps can be determined using the class proportions and the reliability maps by numerical integration of the product of a density ratio and mixture density across the line. We choose logistic regression as our base model for the synthetic task, as it is used often for probability estimation and has linear contours.

For a 5-class problem there are 15 different ways to split the classes into two groups, which correspond to 5 one-vs-four (one-vs-rest) and 10 two-vs-three (pair-vs-rest) tasks. We first generate $n = 400$ (and repeat the same with 2000 and 10000) training instances and train a logistic regression model $g_l$ for each of these tasks, $l = 1, 2, \ldots, 15$. Now we can calculate the true calibration maps $\mu_{g_l}$ and reliability maps $r_{g_l}$. For the model comparing classes 3 and 5 against others these are plotted in Fig. 3. We next learn the calibration maps $\hat{\mu}_{g_l}$ and reliability maps $\hat{r}_{g_l}$ using our method described in previous sections. For the clustering method we use super-cluster size 200 and cluster size 10. For the regression task in learning calibration maps we test bandwidth values 0.005, 0.01, 0.02 and 0.05 and find 0.01 as the best performer. For reliability we use then bandwidth 0.1 as there are $m = 10$ times less instances for training the regression model.

Ultimately we are going to use the estimated calibration and reliability maps for multi-class probability estimation. Therefore, we need the estimated and true distribution of $q(X)$ given $g(X)$ to be maximally alike. As LS-ECOC and LS-ECOC-R both assume Gaussian distribution, we assess how close to each other are $\mathcal{N}(\hat{\mu}_{g_l}, \hat{\sigma}^2_{g_l})$ and $\mathcal{N}(\mu_{g_l}, \sigma^2_{g_l})$, averaged over all instances, where $\sigma^2_{g_l}$ and $\hat{\sigma}^2_{g_l}$ can be calculated as in (8) from the true and estimated calibration and reliability maps, respectively. As a distance measure we use Cramér distance, which is half of the energy distance [14]. Intuitively, it measures how much work has to be done carrying pixels from one density plot to another. The advantage over Kullback-Leibler divergence is that for equal variance it

measures the distance of means. Therefore its value is easier to interpret and perhaps ultimately more relevant for multi-class probability estimation and classification.

With few training instances the regression learner for the reliability map can have a large variance. Therefore, we consider in addition to our local linear regression method an averaging regression method which produces constant reliability across all values of $s$. The method calculates the average of $R^{(m)}$ over all training instances. Table 1 shows the results of the comparison between the estimated and true calibration and probability maps for various training set sizes and the two regression methods. Cramér distance is averaged over results on 10 independently generated training and test sets, the test set size is 10000. For bandwidths we use 0.01 for calibration and 0.1 for reliability, as these are the best according to the results shown later. The results indicate that we are able to learn reliability maps that are better than the average line for already a training set of size 400.

Next we proceed to evaluation of multi-class probabilities that can be obtained using LS-ECOC-R from the learned calibration and reliability maps. For evaluating the probability estimates we use the following measures: root mean square error (RMSE), mean absolute error (MAE), Pearson correlation (Pearson) and Brier score (Brier). For evaluating classification performance we use Error rate (Error) and Error compared to the Bayes-optimal class (ErrVsOpt). Zhou et al. have recently compared 9 ECOC decoding algorithms with regards to classification performance and LS-ECOC out-performed other methods for 7 out of 8 datasets [18]. We use the same datasets (leaving out the two smallest), therefore we compare LS-ECOC-R only against LS-ECOC.

On the synthetic dataset we consider three coding matrices — 'one-vs-rest' (5 columns), 'pair-vs-rest' (10 columns) and 'all' (15 columns). First we study which bandwidth is best for calibration, in order to ensure that we use LS-ECOC at its best. The RMSE between the true and LS-ECOC estimates of multi-class probabilities is presented in Table 2 for three matrices and training set sizes 400, 2000, 10000. The re-

**Table 1.** Comparison of the standard (REL) and averaging (R-AVE) regression method for learning reliabilities, assessed by Cramér distance and averaged over 10 runs. Bandwidths for calibration and regression are 0.01 and 0.1, respectively.

| $n$ | method | 1vsR | 2vsR | 3vsR | 4vsR | 5vsR | 12vsR | 13vsR | 14vsR | 15vsR | 23vsR | 24vsR | 25vsR | 34vsR | 35vsR | 45vsR |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 400 | R-AVE | .0208 | .0253 | .0271 | .0589 | .0567 | .0321 | .0336 | .0577 | .0483 | .0386 | .0678 | .0453 | .0430 | .0684 | .0358 |
| 400 | REL | .0175 | .0233 | .0258 | .0588 | .0564 | .0303 | .0327 | .0564 | .0481 | .0382 | .0672 | .0451 | .0431 | .0683 | .0354 |
| 2000 | R-AVE | .0081 | .0080 | .0093 | .0209 | .0162 | .0101 | .0102 | .0165 | .0137 | .0143 | .0242 | .0127 | .0096 | .0228 | .0096 |
| 2000 | REL | .0046 | .0063 | .0079 | .0203 | .0160 | .0087 | .0097 | .0159 | .0136 | .0139 | .0235 | .0128 | .0095 | .0227 | .0095 |
| 10000 | R-AVE | .0049 | .0038 | .0046 | .0076 | .0043 | .0036 | .0029 | .0054 | .0034 | .0108 | .0072 | .0039 | .0031 | .0073 | .0025 |
| 10000 | REL | .0018 | .0023 | .0035 | .0071 | .0041 | .0025 | .0025 | .0048 | .0032 | .0103 | .0067 | .0037 | .0027 | .0072 | .0024 |

**Table 2.** RMSE of multi-class probability estimates obtained with LS-ECOC on three different ECOC matrices and training set sizes 400, 2000, 10000. Results are averaged over 10 runs.

| matrix | 1vsR | | | 2vsR | | | all | | |
|---|---|---|---|---|---|---|---|---|---|
| $\lambda_{cal}$ $n$ | 400 | 2000 | 10000 | 400 | 2000 | 10000 | 400 | 2000 | 10000 |
| .005 | .1933 | .1621 | .1563 | .1643 | .1478 | .1449 | .1567 | .1421 | .1392 |
| .01 | .1729 | .1594 | .1562 | .1563 | .1480 | .1464 | .1494 | .1425 | .1406 |
| .02 | .1656 | .1597 | .1578 | .1566 | .1515 | .1507 | .1503 | .1461 | .1450 |
| .05 | .1708 | .1673 | .1667 | .1664 | .1635 | .1632 | .1616 | .1588 | .1584 |

sults are averaged over 10 runs over independently generated datasets. Relying on these results we have decided to use bandwidth 0.01 for calibration throughout the paper. As reliability maps are learned on 10 times fewer values because of the clustering, we have chosen 0.1 as the bandwidth for learning reliability.

Table 3 compares the performance of LS-ECOC, LS-ECOC-R-ave and LS-ECOC-R across three matrices and training set sizes 400, 2000, 10000. Both RMSE and error rate are averaged over 10 runs. We also provide the standard deviation estimates for these values, calculated as the sample standard deviation over square root of 10 (the number of runs). The results indicate that both LS-ECOC-R methods outperform LS-ECOC on all cases with $n \geq 2000$ and in some cases with $n = 400$. The full version of LS-ECOC-R performs better than the averaged, with some exceptions for $n = 400$.

It is in principle possible to improve LS-ECOC by improving the calibration map learning method. The following results show that LS-ECOC-R remains superior, with sufficient data given. For this we apply LS-ECOC on the true calibration map (not the estimated one) and consider this as the performance-bound for LS-ECOC. Table 4 compares the results of LS-ECOC, LS-ECOC-R-ave and LS-ECOC-R with the results obtained on the true calibration and reliability maps, averaged over 10 runs. The results indicate that each following method is better than the upper bound for the previous method. Therefore, on this dataset LS-ECOC-R remains superior to LS-ECOC even if calibration is perfect.

**Table 3.** Comparison of LS-ECOC, LS-ECOC-R-ave and LS-ECOC-R across three matrices and three training set sizes. RMSE and error rate are both averaged over 10 runs and standard deviation is calculated as the sample standard deviation over square root of 10.

| $n$ | method | RMSE 1vsR | RMSE 2vsR | RMSE all | Error 1vsR | Error 2vsR | Error all |
|---|---|---|---|---|---|---|---|
| 400 | LS-ECOC | .1729 ± .0015 | .1563 ± .0012 | .1494 ± .0009 | .3860 ± .0065 | .3450 ± .0049 | .3265 ± .0044 |
| 400 | LS-ECOC-R-ave | .1774 ± .0023 | .1526 ± .0020 | .1456 ± .0025 | .3872 ± .0056 | .3572 ± .0051 | .3366 ± .0049 |
| 400 | LS-ECOC-R | .1788 ± .0025 | .1527 ± .0021 | .1475 ± .0025 | .3852 ± .0058 | .3464 ± .0055 | .3325 ± .0059 |
| 2000 | LS-ECOC | .1594 ± .0006 | .1480 ± .0004 | .1425 ± .0004 | .3588 ± .0032 | .3392 ± .0027 | .3184 ± .0031 |
| 2000 | LS-ECOC-R-ave | .1487 ± .0006 | .1314 ± .0005 | .1135 ± .0011 | .3539 ± .0029 | .3367 ± .0025 | .3034 ± .0030 |
| 2000 | LS-ECOC-R | .1468 ± .0006 | .1243 ± .0007 | .1102 ± .0011 | .3493 ± .0026 | .3224 ± .0030 | .3004 ± .0030 |
| 10000 | LS-ECOC | .1562 ± .0004 | .1464 ± .0003 | .1406 ± .0003 | .3509 ± .0020 | .3350 ± .0017 | .3143 ± .0018 |
| 10000 | LS-ECOC-R-ave | .1448 ± .0005 | .1290 ± .0003 | .1111 ± .0004 | .3490 ± .0018 | .3344 ± .0016 | .2996 ± .0019 |
| 10000 | LS-ECOC-R | .1429 ± .0005 | .1227 ± .0002 | .1079 ± .0003 | .3449 ± .0018 | .3214 ± .0016 | .2987 ± .0017 |

**Table 4.** Comparison of LS-ECOC, LS-ECOC-R-ave and LS-ECOC-R with the respective methods which use true calibration and reliability maps (bound). The results were obtained on the full ECOC matrix 'all', on 10000 training instances, and are averaged over 10 runs.

| method | RMSE | MAE | Pearson | Brier | Error | ErrVsOpt |
|---|---|---|---|---|---|---|
| LS-ECOC | .1406 | .1046 | .8801 | .2499 | .3143 | .1327 |
| LS-ECOC bound | .1356 | .0976 | .8843 | .2463 | .3108 | .1239 |
| LS-ECOC-R-ave | .1111 | .0745 | .9253 | .2314 | .2996 | .0943 |
| LS-ECOC-R-ave bound | .1093 | .0727 | .9269 | .2303 | .2988 | .0929 |
| LS-ECOC-R | .1079 | .0712 | .9316 | .2297 | .2987 | .0914 |
| LS-ECOC-R bound | .1018 | .0666 | .9411 | .2264 | .2928 | .0810 |

### 5.2 Experiments on Real Data

Finally we show that LS-ECOC-R outperforms LS-ECOC on some real datasets. For this purpose we use 6 UCI datasets shown in Table 5. As the binary base model we use logistic regression and support vector machines with polynomial kernel. Zhou et al. used also the polynomial kernel and published the best degree for it according to cross-validation results [18]. We use the same degree, for easier comparison with their work.

There are many possible choices for the ECOC matrix, some even domain-specific, in the sense that each column is chosen based on the performance of the models for the previous columns. In the LS-ECOC experiments of Zhou et al. the equidistant code matrices performed best among the domain-independent matrices for 5 of our 6 datasets [18]. We therefore use equidistant code matrices, which we create using BCH codes. We have built a 15x15 binary matrix which was obtained by creating a BCH code with code-length 15 and aligning all 15 code-words with exactly seven 1's as columns of the matrix. The Hamming distance between each pair of columns and each pair of rows is exactly 8, making it equidistant. Another nice property is that any top $k$ rows with $k \geq 5$ have all columns splitting the classes differently into two groups. For a $k$-class problem we use this matrix for our experiments if $k \geq 5$ and for $k = 4$ we use the matrix with all 7 different splits of classes into two.

We perform 10-fold cross-validation and use error-rate as the evaluation measure. The sample mean and variance of the training set are used to normalize each input feature individually. Before calibration, the scores of SVM are transformed using the standard logistic map. The calibration and reliability maps are learned with bandwidths 0.01 and 0.1, respectively — the same which performed best on the synthetic data.

Table 5 lists the error rate for methods LS-ECOC, LS-ECOC-R-ave, LS-ECOC-R on the 6 datasets with two different base learners. We first note that LS-ECOC results in our experiments are superior to the LS-ECOC results by Zhou et al. [18], probably due to differences in normalization and calibration. To assess the differences between LS-ECOC and the two variants of LS-ECOC-R we have performed significance tests with t-test on confidence level 95%. The stars in the table indicate which errors of LS-ECOC-R

**Table 5.** The comparison of 10-fold cross-validated error rate of classification for LS-ECOC, LS-ECOC-R-ave and LS-ECOC-R on 6 real datasets with $n$ instances, $a$ attributes and $k$ classes. The stars in the table indicate which errors of LS-ECOC-R are significantly lower than the respective error for LS-ECOC according to t-test at confidence level 95%.

| dataset | $n$ | $a$ | $k$ | model | LS-ECOC | LS-ECOC-R-ave | LS-ECOC-R |
|---|---|---|---|---|---|---|---|
| shuttle | 14500 | 9 | 7 | LR | .0383± .0016 | .0259± .0014 * | .0323± .0017 * |
| | | | | SVM | .0914± .0015 | .0888± .0017 | .0859± .0018 * |
| sat | 6435 | 36 | 6 | LR | .1713± .0021 | .1514± .0028 * | .1489± .0027 * |
| | | | | SVM | .1737± .0026 | .1554± .0021 * | .1610± .0026 * |
| page-blocks | 5473 | 10 | 5 | LR | .0453± .0025 | .0420± .0034 | .0426± .0035 |
| | | | | SVM | .0426± .0021 | .0429± .0026 | .0411± .0028 |
| segment | 2310 | 19 | 7 | LR | .0887± .0040 | .0775± .0036 | .0753± .0046 * |
| | | | | SVM | .0987± .0041 | .0788± .0036 * | .0771± .0031 * |
| yeast | 1481 | 8 | 10 | LR | .4327± .0134 | .4279± .0124 | .4314± .0165 |
| | | | | SVM | .4084± .0147 | .4246± .0145 | .4198± .0132 |
| vehicle | 846 | 18 | 4 | LR | .2174± .0088 | .2258± .0085 | .2081± .0136 |
| | | | | SVM | .2316± .0125 | .2375± .0113 | .2553± .0136 |

are significantly lower than the respective error for LS-ECOC. To conclude, LS-ECOC-R outperforms LS-ECOC on four larger datasets ($n \geq 2000$), with significance in 3 out of the 4.

## 6 Related Work

The reliability of model predictions has been studied before but mostly in the context of regression, where it is known as conditional variance estimation [5]. Conformal prediction is a general approach that can be applied to both regression and classification in an on-line setting. It outputs a so-called region prediction which might be a confidence interval (in regression) or a set of possible values (in classification) that contains the true value with a certain level of confidence [13]. This is different from the approach in this paper where we try to assess the uncertainty associated with a point estimate.

In the area of multi-class classification and probability scores, the original error-correcting output codes method is due to [4]. The least-squares method for obtaining multi-class posterior probabilities was developed not much later by [8], but does not appear to be widely known. Better-known is the loss-based decoding method by [1], which takes classifier margins rather than probabilities as input and outputs classes rather than posterior probabilities. A review on combinations of binary classifiers in a multi-class setting is given by [9]. [16] study coding and decoding strategies in ECOC, and also originated the synthetic 5-class data set we used in this paper. The same dataset was used by [15] and [18] to study the behaviour of LS-ECOC.

Calibration of multi-class posterior probabilities is often studied in a cost-sensitive setting [17, 11]. The effect of calibration in classifier combination is studied by [2]. Perhaps closest in spirit to our work in this paper is the work by [6] who propose methods to identify and remove unreliable classifiers in a one-vs-one setting. Also related is the work on neighborhood-based local sensitivity by [3].

## 7 Concluding Remarks

Assessing the reliability of probability scores in classification is clearly an important task if we want to combine scores from different classifiers. If we want to combine two scores of 0.5 and 0.3, say, it makes a difference if one of them is deemed much more reliable than the other. Yet the problem of estimating this reliability in an instance-wise manner appears not to have been widely studied in the machine learning literature. In this paper we present a theoretically well-founded and practically feasible approach to the problem. We demonstrate the quality of the reliability estimates both in comparison with the true values on synthetic data, and in obtaining well-calibrated multi-class probability scores through the improved LS-ECOC-R method.

The paper opens many avenues for further work. We are particularly interested in developing cost models in cost-sensitive classification that can take these reliability estimates into account. Incorporating instance-wise confidence ratings into boosting also appears a fruitful research direction.

**Acknowledgments** We would like to thank Tijl De Bie and Thomas Gärtner for helpful discussions. This work is supported by the REFRAME project granted by the European Coordinated Research on Long-term Challenges in Information and Communication Sciences & Technologies ERA-Net (CHIST-ERA), and funded by the Engineering and Physical Sciences Research Council in the UK under grant EP/K018728/1.

## References

1. Erin Allwein, Robert Schapire, and Yoram Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113–141, 2001.
2. Antonio Bella, Cèsar Ferri, José Hernández-Orallo, and María José Ramírez-Quintana. On the effect of calibration in classifier combination. *Applied Intelligence*, 38(4):566–585, 2012.
3. Paul N. Bennett. Neighborhood-based local sensitivity. In *Machine Learning: ECML 2007*, volume 4701 of *Lecture Notes in Computer Science*. 2007.
4. Thomas Dietterich and Ghulum Bakiri. Solving Multiclass Learning Problems via Error-Correcting Output Codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.
5. Jianqing Fan and Qiwei Yao. Efficient estimation of conditional variance functions in stochastic regression. *Biometrika*, 85(3):645–660, 1998.
6. Mikel Galar, Alberto Fernández, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera. Dynamic classifier selection for one-vs-one strategy. *Pattern Recognition*, 46:3412–3424, 2013.
7. Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009.
8. Eun Bae Kong and Thomas Dietterich. Probability estimation via error-correcting output coding. In *International Conference of Artificial Intelligence and Soft Computing*, Banff, Canada, 1997.
9. Ana Carolina Lorena, André Carvalho, and João Gama. A review on the combination of binary classifiers in multiclass problems. *Artificial Intelligence Review*, 30:19–37, 2009.
10. Allan H Murphy. A new vector partition of the probability score. *Journal of Applied Meteorology*, 12(4):595–600, 1973.
11. Deirdre O'Brien, Maya Gupta, and Robert Gray. Cost-sensitive multi-class classification from probability estimates. In *Proceedings of the 25th International Conference on Machine Learning*, pages 712–719, Helsinki, Finland, 2008.
12. Robert E Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. *Machine learning*, 37(3):297–336, 1999.
13. Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9:371–421, 2008.
14. Gábor J Székely and Maria L Rizzo. A new test for multivariate normality. *Journal of Multivariate Analysis*, 93(1):58–80, 2005.
15. Xiaodan Wang and Jindeng Zhou. Research on the characteristic of the probabilistic outputs via LS-ECOC. In *Eighth International Conference on Fuzzy Systems and Knowledge Discovery*, volume 2, pages 1330–1334. IEEE, 2011.
16. Terry Windeatt and Reza Ghaderi. Coding and decoding strategies for multi-class learning problems. *Information Fusion*, 4(1):11–21, March 2003.
17. Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Eighth ACM SIGKDD international conference*, pages 694–699, New York, USA, 2002. ACM Press.
18. Jin Deng Zhou, Xiao Dan Wang, and Heng Song. Research on the unbiased probability estimation of error-correcting output coding. *Pattern Recognition*, 44(7):1552–1565, July 2011.