Mach Learn (2016) 105:459–485
DOI 10.1007/s10994-016-5576-6

Context-based unsupervised ensemble learning and feature ranking

Erfan Soltanmohammadi[1] · **Mort Naraghi-Pour**[1] · **Mihaela van der Schaar**[2]

Received: 8 November 2015 / Accepted: 27 June 2016 / Published online: 12 July 2016

Abstract In ensemble systems, several experts, which may have access to possibly different data, make decisions which are then fused by a combiner (meta-learner) to obtain a final result. Such ensemble-based systems are well-suited for processing big-data from sources such as social media, in-stream monitoring systems, networks, and markets, and provide more accurate results than single expert systems. However, most existing ensemble-learning techniques have two limitations: (i) they are supervised, and hence they require access to the true label, which is often unknown in practice, and (ii) they are not able to evaluate the impact of the various data features/contexts on the final decision, and hence they do not learn which data is required. In this paper we propose a joint estimation–detection method for evaluating the accuracy of each expert as a function of the data features/context and for fusing the experts decisions. The proposed method is unsupervised: the true labels are not available and no prior information is assumed regarding the performance of each expert. Extensive simulation results show the improvement of the proposed method as compared to the state-of-the-art approaches. We also provide a systematic, unsupervised method for ranking the informativeness of each feature on the decision making process.

Keywords Ensemble learning · Unsupervised learning · Decision making · Contextual estimation · Feature selection · Big data

Editor: Johannes Fürnkranz.

✉ Mort Naraghi-Pour
naraghi@lsu.edu

Erfan Soltanmohammadi
erfan.soltanmohammadi@kla-tencor.com

Mihaela van der Schaar
mihaela@ee.ucla.edu

[1] School of Electrical Engineering and Computer Science, Louisiana State University, Baton Rouge, LA 70803, USA

[2] Electrical Engineering Department, University of California, Los Angeles, Los Angeles, CA 90095, USA

1 Introduction

In numerous big data applications [e.g., data-driven marketing (Brown et al. 2011), surveillance (Craig and Ludloff 2011), sensing and networking (Segaran and Hammerbacher 2009), and health monitoring (Tseng et al. 2008)] involving data mining, decision making, predictions etc., ensemble-based approaches have been shown to produce more accurate results than single-expert systems (Kuncheva and Whitaker 2003; Tekin and van der Schaar 2013; Zhang et al. 2013). Another key advantage is that, when the various experts have access to and base their decisions on heterogeneous sources of data, ensemble-based approaches do not need to centralize the data acquisition and processing, thereby enabling low-delay, distributed processing by individual experts.

An ensemble system is constructed from a set of (possibly heterogeneous)[1] experts and a proper combining rule for fusing the outputs of the experts. Individual experts may have access to heterogeneous data and may have been trained using different data sets. Hence, by properly combining their outputs, ensemble-based methods can achieve more accurate decisions.

As mentioned above, in many applications, the data may be distributed among the experts, with each of the experts using a part of the data. The data may be partitioned horizontally so that each expert works with different disjoint subsets of the entire data set, or vertically so that each expert works with a subset of dimensions (or features) of the same data (Zhang et al. 2013; Zheng et al. 2011).

It is well-known that the success of ensemble methods depends on the diversity of the experts. Bagging (Bootstrap aggregating, Breiman 1996), Boosting, (Schapire 1990), and AdaBoost (Freund and Schapire 1997) represent examples of ensemble learning methods in which diversity is achieved by using different training subsets. Neural networks and decision trees represent examples in which diversity is achieved based on the structure of the expert and the parameters selected during their training stage.

In any ensemble system, the combiner, which combines the local decisions of the experts, plays an essential role in determining the overall performance. Different methods have been proposed to aggregate the individual decisions of experts. When the performance of experts is unknown, majority rule is often employed (Kuncheva 2004), while when the performances of the experts is known, weighted majority rules are often employed, in which different optimally-designed weights are assigned to the experts based on their accuracies.

The method of tracking the best expert is one of the seminal works in online ensemble learning based on weighted majority rule (Herbster and Warmuth 1998). In this approach, the importance of each expert is modeled by a weight which is updated over time using an adaptation method. Different variations of this method have been proposed in which the fusion rule or adaptation algorithm were improved. To improve the adaptation equation, new cost functions are suggested in Choromanska and Monteleoni (2012), Herbster and Warmuth (2001), Monteleoni and Jaakkola (2004).

A priori information regarding the performance of each expert can be obtained using training and validation data sets. For instance, the behavior knowledge space (BKS) method estimates the densities of the classifier outputs and requires large training and validation data sets (Huang and Suen 1995). In some ensemble systems, the experts and the combiner are trained together, using a joint procedure, such as stacked generalization or mixture of experts (Jacobs et al. 1991; Wolpert 1992).

[1] Here heterogeneity of classifiers implies that they may adopt different processing schemes, which may lead to different error rates in classifying the data (Webb and Copsey 2011).

Optimal fusion of local decisions requires the a priori knowledge of the accuracy of the experts which, in many applications, may not be available. For example, the data may have an extremely large dimension or the data stream may be time-varying which makes it difficult to accurately evaluate the experts' performance based on a priori, limited validation data sets. Moreover, data streams are often received along with their context. The context could be a small side information such as a description of the way the data is acquired (Tekin and van der Schaar 2013), or it could be a small dimensional portion of the actual high dimensional data representing one of its features or attributes. However, the accuracies of the experts often vary with the context, and the combiner needs to know the accuracies of the experts for every arriving context in order to optimally fuse their decisions, resulting in prohibitively high cost in processing, communication and storage requirements.

In this paper we present an unsupervised ensemble learning method in which the combiner has no prior information regarding the experts' performance. In addition, the methods adopted by the experts or the data in which they operate is also unknown by the combiner. Each expert may use a different part of the big data, the preprocessed data, or even different correlated data streams obtained from multiple sources. The combiner uses an unsupervised approach to evaluate the accuracies of the experts as functions of the data context as well as to fuse the decisions of individual experts. We introduce a model for estimating the experts' accuracies in terms of probabilities of false alarm and correct detection.

To contrast our approach with those in Choromanska and Monteleoni (2012), Herbster and Warmuth (1998), Herbster and Warmuth (2001), Monteleoni and Jaakkola (2004), we would like to point out that the main focus of these papers is to design an *online* fusion rule using the unsupervised weighted majority rule. On the other hand, our approach uses batch processing. We assume that the data is received along with some context and that the performance of the individual experts is unknown. Our proposed method estimates the performance of the experts in terms of their probabilities of detection and false alarm as a function of the data context, and fuses the decisions of the individual experts. A novel feature of our approach is the manner in which we develop the expectation maximization (EM) algorithm to enable ensemble learning. Ordinarily, for a set of I instances or time indexes, the well-known EM algorithm (Dempster et al. 1977) must run for 2^I runs in order to obtain an estimation of the parameters and the fused decisions for the I instances. Instead, we introduce separate prior probabilities for the fused decision of each instance. This allows us to obtain an estimate of the parameters and the fused decision for the I instances from a single run of the EM algorithm. We show that, even though unsupervised, our proposed ensemble learning method outperforms numerous state-of-the-art ensemble approaches that are supervised.

In many applications we wish to determine the importance or influence of different features on the final decision. Previously, different traditional feature selection methods have been proposed in Holte (1993), Karegowda et al. (2010), Roobaert et al. (2006), Kannan and Ramaraj (2010). These are supervised methods in which the true labels are known, and the features are selected based on different criteria. Mutual information quotient (MIQ) and mutual information difference (MID) are two effective feature selection methods which are based on the mutual information between the true label and different features (Ding and Peng 2003; Peng et al. 2005). The main drawback of these methods is that they are supervised, i.e., they need to know the true label. We extend our proposed method to select/rank the features (data contexts), in terms of their impact on the ensemble's decision making process. We show that, even though unsupervised, our proposed feature selection method is similar in performance to supervised feature selection methods such as MIQ and MID.

A preliminary version of this paper appears in the proceedings of the 32nd International Conference on Machine Learning, pp. 2076-2084, 2015. The current manuscript has the following changes/additions from our ICML paper.

- We have explained our algorithm in more detail, with derivations and with additional discussions.
- We have provided a discussion on the computational complexity of the proposed algorithm.
- Assuming that the expectation maximization algorithm converges to the maximum likelihood solution, we have justified the combiner's fusion rule.
- Using an information theoretic approach, we have proposed a new feature ranking algorithm. The results show that this new feature ranking approach provides a performance similar to the supervised ranking methods.
- We have included several additional results on the performance of the algorithm, on a comparison of the proposed method with the majority rule, and on the effect of the values of the parameters of the algorithm on its performance.

The rest of this paper is organized as follows. In Sect. 2, we introduce the required notations and formulate the problem. In Sect. 3, the proposed approach is developed. The feature selection procedure is introduced in Sect. 4. Finally, numerical results and concluding remarks are presented in Sects. 5 and 6, respectively.

2 Problem formulation and notations

We consider an ensemble learning system with K experts; each expert classifies an input data stream characterized by its context.

Since a multiple-choice decision making problem can be divided into a set of binary decision problems (Lienhart et al. 2003), without loss of generality we consider the binary decision problem here.

For each instance[2] i, let the portion of data available for the kth expert be denoted by $s_k(i) \in \mathcal{S}_k$, and let $Z(i) \in \mathcal{Z}$ be the context of the received data. As mentioned before, the context may be a vector in general, and may represent a side information about the data or it may be a subset of the features (attributes) of the data. The set $\mathcal{Z}$ is assumed to be a (subset of a) metric space with the metric $d_{\mathcal{Z}}(z_1, z_2)$ that represents the distance between z_1 and z_2. Let $y(i) \in \mathcal{Y} \triangleq \{0, 1\}$ denote the true label at instance i. In the proposed approach, the true label $y(i)$ is not available to the combiner/ensemble learner and the combiner does not know the methods used by the experts to classify the data. Our unsupervised method will use the context $Z(i)$ to estimate the accuracy of each expert.

Let $\mathbf{Z} \triangleq [Z(1)\ Z(2)\ \ldots\ Z(I)]$ and $\mathbf{y} \triangleq [y(1)\ y(2)\ \ldots\ y(I)]$ denote the observed vector of contexts and the unobserved vector of true labels, respectively, for a duration I starting at 1. As mentioned previously, $\mathbf{y}$ is not available and its detection is also a part of the proposed approach. We define the *label matrix*, Δ, by

$$\Delta \triangleq \begin{bmatrix} \delta_0(1) & \delta_0(2) & \cdots & \delta_0(I) \\ \delta_1(1) & \delta_1(2) & \cdots & \delta_1(I) \end{bmatrix} \tag{1}$$

where column i corresponds to the true label $y(i)$, and at each instance i, one of the elements in column i is 1 and the other is 0. If $\delta_0(i) = 0$, then $\delta_1(i) = 1$, indicating that at instance

[2] For other applications such as processing a database, instance can be replaced by the index of the data sample.

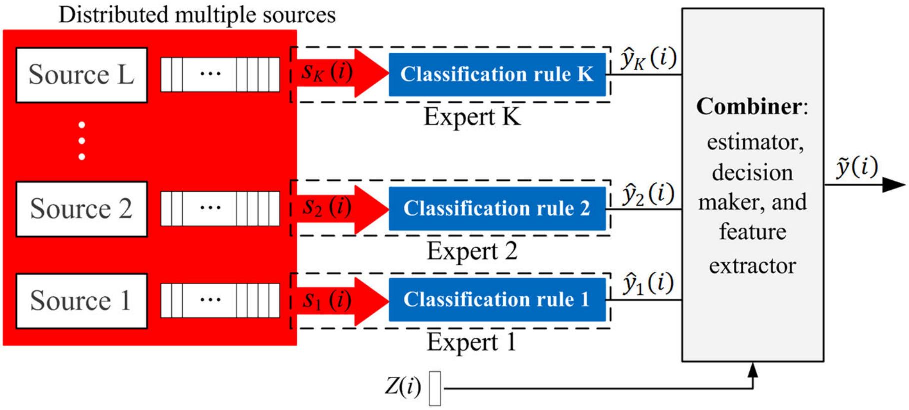

Fig. 1 System model includes data $s_k(i)$ for $k = 1, 2, \ldots, K$ received from distributed multiple sources at instance i to K experts for making local decisions. The context $Z(i)$ of data is also available to the expert system. The combiner (learner) uses the local decisions from the experts and the contexts from the sources in order to estimate the accuracy of each expert as a function of its context and also to fuse the local decisions to make the final decision, $\tilde{y}(i)$. The combiner will also extract the importance of different features of the data in the decision making process

i we have $y(i) = 1$; similarly, if $\delta_0(i) = 1$, then $\delta_1(i) = 0$, indicating that at instance i we have $y(i) = 0$.

Let $\hat{y}_k(i)$ be the local decision of the kth expert at instance i and let $\hat{\mathbf{y}}(i) = \left[\hat{y}_1(i)\ \hat{y}_2(i)\ \ldots\ \hat{y}_K(i)\right]^\dagger$ denote the vector of K local decisions at instance i, where † represents the transpose operation. Finally, let Y denote the collection of local decisions for duration I defined by the following matrix.

$$Y \triangleq \begin{bmatrix} \hat{y}_1(1) & \hat{y}_1(2) & \cdots & \hat{y}_1(I) \\ \hat{y}_2(1) & \hat{y}_2(2) & \cdots & \hat{y}_2(I) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{y}_K(1) & \hat{y}_K(2) & \cdots & \hat{y}_K(I) \end{bmatrix} \tag{2}$$

The entire system is shown in Fig. 1. The system is comprised of a set of diverse experts. Every expert makes a local decision which it delivers to the combiner for the final decision. Individual experts may be trained with different data sets and may have different error rates. As shown Fig. 1, the performance of an expert is affected by the part of data dedicated to it and its classification rule. In the rest of the paper, in order to make the problem mathematically tractable, we assume that given the label and context, the decisions of the individual experts are independent.

The combiner receives the decisions of all the experts, Y, (as well as the context $\mathbf{Z}$) and needs to fuse them to get an estimate of the (unknown) true labels. However, to enable the efficient fusion of the received decisions, the combiner must estimate the accuracy of each expert. We describe these accuracies in terms of the probabilities of correct decision for each expert. More specifically, we associate a probability of (correct) detection[3] and a probability of false alarm[4] with each expert. In order to estimate these probabilities, we require the true labels (which are unknown). On the other hand, (for the combiner) to detect the true labels, we

[3] The probability that the expert makes a correct determination of the true label when the label is 1.

[4] The probability that the expert makes a an incorrect determination of the true label when the label is 0.

require the probabilities of detection and false alarm. From this, it can be easily seen that these two problems are connected. A naive solution is to estimate the probabilities of false alarm and detection for every possible label (decision) vector from 1 to I (2^I possibilities), and then use these estimated probabilities to evaluate the likelihood of observing the corresponding label vector, and among all the label vectors, select the one with the highest likelihood. Clearly, the computational complexity of this approach is prohibitive. In the next section we present a novel method based on the EM algorithm which can be used to effectively detect the true labels and estimate the probabilities of false alarm and detection for each expert with significantly lower complexity than the brute force method.

In addition to characterizing the accuracy of each expert, the probabilities of detection and false alarm model the changes in the input statistics of each expert. Furthermore, we note that the effect of noisy data on the performance of each expert can also be included in these probabilities. In particular, higher noise levels in the data result in a lower performance for the experts in terms of probabilities of false alarm and detection. Since the performance of an expert is determined by the context of its acquired data, these probabilities are assumed to be functions of the context. For a fixed context, however, an expert has fixed probabilities of detection and false alarm. Therefore, for context z and for expert k, we define the probability of detection, denoted by $p_{1k}(z)$, and the probability of false alarm, denoted by $p_{0k}(z)$ as

$$p_{\eta k}(z) \triangleq p\left(\hat{y}_k(i) = 1 \mid \delta_\eta(i) = 1; z\right), \quad \eta = 0, 1 \tag{3}$$

We assume that these probabilities are Lipschitz continuous with Lipschitz constant $c_{\eta k}$, i.e.,

$$|p_{\eta k}(z_1) - p_{\eta k}(z_2)| \;\leq\; c_{\eta k}\, d_{\mathcal{Z}}\,(z_1, z_2) \tag{4}$$

where $d_{\mathcal{Z}}$, defined previously, is the metric on the set $\mathcal{Z}$. This assumption, which imposes a constraint on how fast an expert's accuracy can change with context, is clearly valid in most practical situations (Kleinberg et al. 2008; Tekin and van der Schaar 2013). We arrange these probabilities for all the experts into a matrix $P(z) \triangleq [p_{\eta k}(z)]$, $\eta = 0, 1$, $k = 1, 2, \ldots, K$. Note that the combiner does not know $P(z)$ and one of the goals of our proposed method is to estimate it.

We assign prior probabilities $\phi_0(i)$ and $\phi_1(i)$ to the true label $y(i)$ for $i = 1, 2, \ldots, I$ and arrange them in a matrix as follows[5]

$$\Phi \triangleq \begin{bmatrix} \phi_0(1) & \phi_0(2) & \cdots & \phi_0(I) \\ \phi_1(1) & \phi_1(2) & \cdots & \phi_1(I) \end{bmatrix} \tag{5}$$

where $\phi_\eta(i) = p(\delta_\eta(i) = 1)$ and $\phi_0(i) + \phi_1(i) = 1$. Please note that neither Δ nor Φ are available to the combiner. They are assumed to be unknown parameters which are evaluated in the proposed method in order to estimate P and to detect $\mathbf{y}$. To summarize, the two-tuple, $\Theta = \{P(z), \Phi\}$ is defined as the unknown *parameter set* which the combiner tries to estimate based on the local decisions of the experts, Y, and context of the data, $\mathbf{Z}$. After estimating the parameter set Θ, the combiner detects the true labels $\mathbf{y}$. In the next section, we propose an approach based on the EM algorithm for the combiner to achieve these goals. A timing diagram of available and unavailable information is shown in Fig. 2.

[5] We would like to point out that although we refer to Φ as the prior probability matrix, it is only introduced here to convert the problem of detection of $\mathbf{y}$ into a problem of estimation of the matrix Φ. This point is made clear in Sect. 3.2.

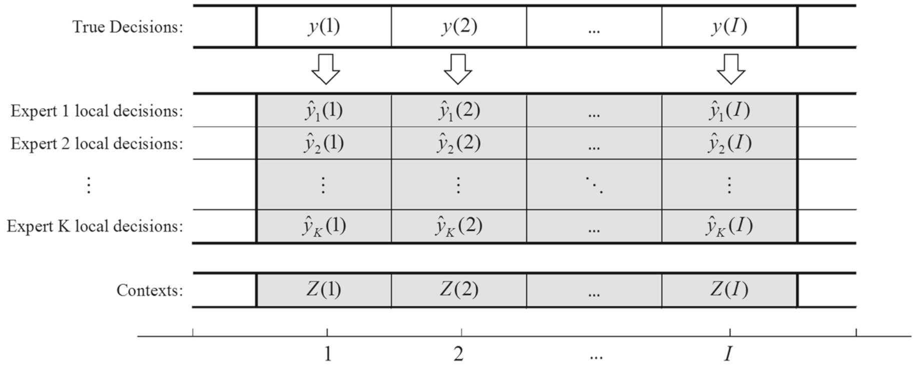

Fig. 2 Timing diagram of the proposed system. The *gray areas* include Y and $\mathbf{Z}$ show the available information for the combiner. Note that the true labels in the *first row* are not available

3 Estimation of the experts' accuracies and decision making

In this section, given the local decisions, Y, and the observed vector of contexts, $\mathbf{Z}$, we first develop an estimation method for Θ (which includes the estimation of $P(z)$, $\forall z \in \mathcal{Z}$, and Φ). Then, we use the estimated parameters to detect the true labels $\mathbf{y}$.

3.1 Estimation procedure

The maximum likelihood estimate of Θ given Y and $\mathbf{Z}$ is given by

$$\hat{\Theta} = \arg\max_{\Theta} p\left(Y|\Theta, \mathbf{Z}\right) = \arg\max_{\Theta} \sum_{\Delta} p\left(Y, \Delta \mid \Theta, \mathbf{Z}\right) \tag{6}$$

By considering Δ as a latent variable, the mixture model in (6) can be iteratively solved using the EM algorithm. First, we evaluate $p(Y, \Delta|\Theta, \mathbf{Z})$ by

$$p\left(Y, \Delta|\Theta, \mathbf{Z}\right) = p\left(Y|\Delta; \Theta, \mathbf{Z}\right) p\left(\Delta|\Theta, \mathbf{Z}\right) \tag{7}$$

where $p(Y|\Delta; \Theta, \mathbf{Z})$ represents the probability that the K experts, over the I instances, make the decisions arranged as the matrix Y, given that: the parameter set, Θ is known, the contexts are as in $\mathbf{Z}$, and the actual label matrix is Δ. Given this condition,[6] observations $\hat{y}_k(i)$ and $\hat{y}_m(j)$ will be independent for $m \neq k$ or $i \neq j$. Also since the actual label is independent of experts parameters and the context, i.e., $p(\Delta|\Theta, \mathbf{Z}) = p(\Delta)$. Here we assume the actual labels are independent over the instances. Therefore,

$$p\left(Y, \Delta|\Theta, \mathbf{Z}\right) = \prod_{k}\prod_{i}\prod_{\eta}\left[p_{\eta k}^{\hat{y}_k(i)}(Z(i))\left(1 - p_{\eta k}(Z(i))\right)^{1-\hat{y}_k(i)}\phi_{\eta}^{\frac{1}{K}}(i)\right]^{\delta_{\eta}(i)} \tag{8}$$

Note that in this section and for all the products and summations, i is in the range 1 to I, k goes from 1 to K, and η is in $\{0,\ 1\}$. The log-likelihood function is obtained as

[6] As mentioned previously, we assume that given the true labels, i.e., Δ, and the contexts Z (as well as the parameter set Θ), the experts' decisions are independent.

$$
\begin{aligned}
L\left(\Theta; Y, \Delta, \mathbf{Z}\right) &= \log p\left(Y, \Delta \mid \Theta, \mathbf{Z}\right) \\
&= \sum_k \sum_i \sum_\eta \delta_\eta(i) \Big[\hat{y}_k(i) \log p_{\eta k}(Z(i)) \\
&\quad + (1 - \hat{y}_k(i)) \log \left(1 - p_{\eta k}(Z(i))\right) + \frac{1}{K} \log \phi_\eta(i) \Big] \qquad (9)
\end{aligned}
$$

After finding the log-likelihood function we are able to construct the two steps of EM algorithm: the expectation and the maximization steps. They are described below.

Expectation step

In this step, the expectation of the log-likelihood function, denoted by $Q(\Theta; \Theta^{\text{old}})$ is evaluated with respect to the conditional distribution $p(\Delta \mid Y; \Theta^{\text{old}})$ of the latent variable Δ, where Θ^{old} is the previous estimate for Θ. That is,

$$
\begin{aligned}
Q\left(\Theta; \Theta^{\text{old}}\right) &= E_{\Delta \mid Y; \Theta^{\text{old}}} \left[L(\Theta; Y, \Delta, \mathbf{Z})\right] \\
&= \sum_k \sum_i \sum_\eta \gamma(\eta, i) \Big[\hat{y}_k(i) \log p_{\eta k}(Z(i)) \\
&\quad + \left(1 - \hat{y}_k(i)\right) \log \left(1 - p_{\eta k}(Z(i))\right) + \frac{1}{K} \log \phi_\eta(i) \Big] \qquad (10)
\end{aligned}
$$

where $E_{A \mid C, D, \ldots}$ denotes expectation with respect to A given the variables C and $D, \ldots$, and where

$$
\begin{aligned}
\gamma(\eta, i) &= E_{\Delta \mid Y; \Theta^{\text{old}}} \left[\delta_\eta(i)\right] = p(\delta_\eta(i) = 1 \mid Y; \Theta^{\text{old}}, Z(i)) \\
&= p(\delta_\eta(i) = 1 \mid \hat{\mathbf{y}}(i); \Theta^{\text{old}}, Z(i)) \\
&= \frac{p(\hat{\mathbf{y}}(i) \mid \delta_\eta(i) = 1; \Theta^{\text{old}}, Z(i)) p(\delta_\eta(i) = 1 \mid \Theta^{\text{old}}, Z(i))}{\sum_{j=0}^{1} p(\hat{\mathbf{y}}(i) \mid \delta_{jt} = 1; \Theta^{\text{old}}, Z(i)) p(\delta_{jt} = 1 \mid \Theta^{\text{old}}, Z(i))} \\
&= \frac{\phi_\eta^{\text{old}}(i) \prod_k \left(p_{\eta k}^{\text{old}}(Z(i))\right)^{y_k(i)} \left(1 - p_{\eta k}^{\text{old}}(Z(i))\right)^{1 - y_k(i)}}{\sum_{j=0}^{1} \phi_j^{\text{old}}(i) \prod_k \left(p_{jk}^{\text{old}}(Z(i))\right)^{y_k(i)} \left(1 - p_{jk}^{\text{old}}(Z(i))\right)^{1 - y_k(i)}} \\
&= \frac{\phi_\eta^{\text{old}}(i) \prod_k \left(p_{\eta k}^{\text{old}}(Z(i))\right)^{y_k(i)} \left(1 - p_{\eta k}^{\text{old}}(Z(i))\right)^{1 - y_k(i)}}{\sum_{j=0}^{1} \phi_j^{\text{old}}(i) \prod_k \left(p_{jk}^{\text{old}}(Z(i))\right)^{y_k(i)} \left(1 - p_{jk}^{\text{old}}(Z(i))\right)^{1 - y_k(i)}} \qquad (11)
\end{aligned}
$$

Maximization step

In this step, $Q(\Theta; \Theta^{\text{old}})$ is maximized with respect to Θ. In maximizing $Q(\Theta; \Theta^{\text{old}})$ with respect to $\phi_\eta(i)$ we must consider the constraint $\sum_{\eta=0}^{1} \phi_\eta(i) = 1$. Therefore using the Lagrange multiplier method, we maximize the Lagrangian $\breve{Q}\left(\Theta; \Theta^{\text{old}}, \lambda_i\right)$ given by

$$
\breve{Q}\left(\Theta; \Theta^{\text{old}}, \lambda_i\right) = Q(\Theta; \Theta^{\text{old}}) + \lambda_i \left\{ \sum_{\eta=0}^{1} \phi_\eta(i) - 1 \right\} \qquad (12)
$$

which gives

$$
\frac{\partial \breve{Q}}{\partial \phi_\eta(i)} = \frac{\gamma(\eta, i)}{\phi_\eta(i)} + \lambda_i = 0 \qquad (13)
$$

Multiplying both sides by $\phi_\eta(i)$ and summing over η gives $\lambda_i = -\sum_{j=0}^{1}\gamma(j,i)$, which results in

$$\phi_\eta^{\text{new}}(i) = \frac{\gamma(\eta, i)}{\sum_{j=0}^{1}\gamma(j,i)} = \gamma(\eta, i) \tag{14}$$

We would like to note that since $Q(\Theta;\Theta^{\text{old}})$ is a concave function of $\phi_\eta(i)$, and the constraint is linear, the solution of the above Lagrangian in fact maximizes $Q(\Theta;\Theta^{\text{old}})$.

Maximization of $Q(\Theta;\Theta^{\text{old}})$ with respect to $p_{\eta k}(Z(i))$ is also a constraint optimization problem given by

$$\begin{aligned} &p_{\eta k}^{\text{new}}(Z(i)) = \underset{p_{\eta k}(Z(i))}{\arg\max}\; Q\left(\Theta;\Theta^{\text{old}}\right) \\ &\quad\text{subject to:} \\ &\quad|p_{\eta k}(z_1) - p_{\eta k}(z_2)| \le c_{\eta k} d_{\mathcal{Z}}(z_1, z_2),\ \forall z_1, z_2 \in \mathcal{Z} \\ &\quad 0 \le p_{\eta k}(z) \le 1 \text{ for } \eta = 0, 1,\ k = 1, 2, \ldots, K,\ \forall z \in \mathcal{Z} \end{aligned} \tag{15}$$

Since log(.) is a concave function and $\gamma(\eta, i)$ and $\hat{y}_k(i)$ are non-negative, and since non-negative weighted sum of concave functions is still concave, $Q(\Theta;\Theta^{\text{old}})$ is a concave function of $p_{\eta k}(Z(i))$. Therefore we can use convex optimization approaches to solve (15). Towards this let

$$\varrho_{\eta k}(l, j) \triangleq c_{\eta k} d_{\mathcal{Z}}(z(l), z(j)) \tag{16}$$

$$\mathbf{p}_{\eta k} \triangleq [p_{\eta k}(Z(1)), p_{\eta k}(Z(2)), \ldots, p_{\eta k}(Z(I))]^{\dagger} \tag{17}$$

$$\psi(\mathbf{p}_{\eta k}) \triangleq \sum_i \gamma(\eta, i)\Big(\hat{y}_k(i)\log p_{\eta k}(Z(i)) + (1-\hat{y}_k(i))\log\left(1 - p_{\eta k}(Z(i))\right)\Big) \tag{18}$$

Then, to maximize $Q(\Theta;\Theta^{\text{old}})$ with respect to $p_{\eta k}(z)$ subject to the Lipschitz continuity constraint in (4), we can solve the constrained optimization problem given by

$$\begin{aligned} &\mathbf{p}_{\eta k}^{\text{new}} = \underset{\mathbf{p}_{\eta k}}{\arg\max}\; \psi(\mathbf{p}_{\eta k}) \\ &\quad\text{subject to:} \\ &\quad|p_{\eta k}(z(l)) - p_{\eta k}(z(j))| \le \varrho_{\eta k}(l, j)\ \forall\, l, j,\ i = 1, 2, \ldots, I, \\ &\quad 0 \le p_{\eta k}(Z(i)) \le 1 \text{ for } \eta = 0, 1,\ k = 1, 2, \ldots, K \end{aligned} \tag{19}$$

We can rewrite (19) as

$$\begin{aligned} &\mathbf{p}_{\eta k}^{\text{new}} = \underset{\mathbf{p}_{\eta k}}{\arg\max}\; \psi(\mathbf{p}_{\eta k}) \\ &\quad\text{subject to: } \Lambda\mathbf{p}_{\eta k} \le \boldsymbol{\varrho} \text{ and } \mathbf{0} \le \mathbf{p}_{\eta k} \le \mathbf{1} \end{aligned} \tag{20}$$

where the inequalities are component-wise, and **0** and **1** are the all-zero and the all-one column vectors of length I, respectively, and where

$$\Lambda \triangleq \begin{array}{c|ccccccc|}
 & 1 & 2 & 3 & 4 & \cdots & I-1 & I \\
1 & 1 & -1 & 0 & 0 & \cdots & 0 & 0 \\
2 & -1 & 1 & 0 & 0 & \cdots & 0 & 0 \\
3 & 1 & 0 & -1 & 0 & \cdots & 0 & 0 \\
4 & -1 & 0 & 1 & 0 & \cdots & 0 & 0 \\
\vdots & & & & & & & \\
2I-1 & 1 & 0 & 0 & 0 & \cdots & 0 & -1 \\
2I & -1 & 0 & 0 & 0 & \cdots & 0 & 1 \\
\cline{2-8}
2I+1 & 0 & 1 & -1 & 0 & \cdots & 0 & 0 \\
2I+2 & 0 & -1 & 1 & 0 & \cdots & 0 & 0 \\
2I+3 & 0 & 1 & 0 & -1 & \cdots & 0 & 0 \\
2I+4 & 0 & -1 & 1 & 1 & \cdots & 0 & 0 \\
\vdots & & & & & & & \\
4I-1 & 1 & 0 & 0 & 0 & \cdots & 0 & -1 \\
4I-2 & -1 & 0 & 0 & 0 & \cdots & 0 & 1 \\
\cline{2-8}
\vdots & & & & \vdots & & & \\
\cline{2-8}
I^2-I-1 & 0 & 0 & 0 & 0 & \cdots & 1 & -1 \\
I^2-I & 0 & 0 & 0 & 0 & \cdots & -1 & 1
\end{array} \tag{21}$$

and

$$\boldsymbol{\varrho}_{\eta k} \triangleq \begin{array}{c|c|}
1 & \varrho_{\eta k}(1,2) \\
2 & \varrho_{\eta k}(1,2) \\
3 & \varrho_{\eta k}(1,3) \\
4 & \varrho_{\eta k}(1,3) \\
\vdots & \vdots \\
2I-1 & \varrho_{\eta k}(1,I) \\
2I & \varrho_{\eta k}(1,I) \\
\cline{2-2}
2I+1 & \varrho_{\eta k}(2,3) \\
2I+2 & \varrho_{\eta k}(2,3) \\
2I+3 & \varrho_{\eta k}(2,4) \\
2I+4 & \varrho_{\eta k}(2,4) \\
\vdots & \vdots \\
4I-1 & \varrho_{\eta k}(2,I) \\
4I-2 & \varrho_{\eta k}(2,I) \\
\cline{2-2}
\vdots & \vdots \\
\cline{2-2}
I^2-I-1 & \varrho_{\eta k}((I-2),I) \\
I^2-I & \varrho_{\eta k}((I-1),I)
\end{array} \tag{22}$$

Here the maximization is performed with respect to the $\mathbf{p}_{\eta k}$, $\eta = 0, 1$, $k = 1, 2, \ldots, K$, and the rest of the parameters are considered fixed. Note that the objective function is concave and the constraints are linear; therefore, (20) can be solved using interior point methods (Boyd and Vandenberghe 2004).

By iterating between the expectation step and the maximization step, until a stopping criterion is satisfied,[7] we find an estimation of the parameter set.[8]

In each iteration of the EM, Eqs. (11), (14), and (20) are calculated. (14) is directly derived from (11), so they together need $\mathcal{O}(KI)$ multiplications. However, in each iteration to solve (14), a use of the interior point algorithm is required which would be the most dominant term in the computational complexity and requires $\mathcal{O}(\sqrt{KI})$ in each of its iterations. Assuming N_{IP} iterations are required for interior point, the computational complexity would be $\mathcal{O}(N_{\text{IP}}\sqrt{KI})$ (Anstreicher 1999). Finally, assuming that we run the EM N_{EM} times, the computational complexity of the entire algorithm would be $\mathcal{O}(N_{\text{EM}}N_{\text{IP}}\sqrt{KI})$.

We denote the final estimates of the parameter set by $\tilde{\Theta}$. Similarly we denote the final estimates of P and Φ and their entries $p_{\eta k}(z)$ and $\phi_\eta(i)$ by $\tilde{P}$, $\tilde{\Phi}$, $\tilde{p}_{\eta k}(z)$ and $\tilde{\phi}_\eta(i)$, respectively.

In order to evaluate $p_{\eta k}(z)$ for all $z \in \mathcal{Z}$, we note that for any $j = 1, 2, \ldots, I$, $\eta = 0, 1$ and $k = 1, 2, \ldots, K$,

$$\tilde{p}_{\eta k}(z(j)) - c_{\eta k} d_{\mathcal{Z}}(z, z(j)) \leq \tilde{p}_{\eta k}(z) \tag{23}$$

$$\tilde{p}_{\eta k}(z(j)) + c_{\eta k} d_{\mathcal{Z}}(z, z(j)) \geq \tilde{p}_{\eta k}(z) \tag{24}$$

Therefore, we can interpolate the values of $p_{\eta k}(z(1+l)), l = 0, 1, \ldots, I-1$ to obtain $p_{\eta k}(z)$ for any $z \in \mathcal{Z}$. Let

$$p1_{\eta k}(z) = \max_{1 \leq j \leq I} \left\{\tilde{p}_{\eta k}(z(j)) - c_{\eta k} d_{\mathcal{Z}}(z, z(j))\right\} \tag{25}$$

and

$$p2_{\eta k}(z) = \min_{1 \leq j \leq I} \left\{\tilde{p}_{\eta k}(z(j)) + c_{\eta k} d_{\mathcal{Z}}(z, z(j))\right\} \tag{26}$$

We then set[9]

$$\tilde{p}_{\eta k}(z) = \min\left\{p1_{\eta k}(z), p2_{\eta k}(z)\right\} \tag{27}$$

Remark 1 The Lipschitz constants $c_{\eta k}$ affect the performance of the algorithm in estimating the parameters $p_{\eta k}(z)$ as functions of z. As evident from (4), (23) and (27), smaller values of $c_{\eta k}$ result in a smoother estimate for $p_{\eta k}(z)$, while larger values of $c_{\eta k}$ allow for larger variations in the estimates. Therefore the Lipschitz constants $c_{\eta k}$ must be selected in accordance with the performance of the classifiers as a function of the context variables. In particular, if for example the detection performance $p_{1k}(z)$ of the kth classifier is believed to be very sensitive to the context variable z, i.e., small changes in z result in large changes in $p_{1k}(z)$, then the value of c_{1k} must be chosen to be large. On the other hand, if the detection performance of the k-th classifier is not very sensitive to the context variable z, then a smaller value should be assigned to c_{1k}. That said, we would like to also point out that the Lipschitz condition in (4) is introduced to enable the estimation of the functions $p_{\eta k}(z)$ with a smaller number

[7] A stopping criterion could be a pre-selected number of iterations or a threshold on the relative difference between the last two estimations. In Sect. 5, we have used the number of iterations as the stopping criterion where we show the results of the parameter estimation for 1, 2 and 5 iterations of the algorithm. It is shown that the estimated parameters after 2 and 5 iterations are very close and also close to the actual parameters.

[8] For a discussion of the convergence properties of the EM algorithm we refer to Dempster et al. (1977), Bishop (2006).

[9] The minimum in (27) provides a maxmin approximation for the values of detection (false alarm) probabilities that have been calculated. This is an interpolation problem and our approach is admittedly heuristic. Another approach is to select the mean.

of samples. It is important to note that even if $c_{\eta k}$ do not satisfy (4) for the true functions $p_{\eta k}(z)$, our algorithm still works. However, in this case our estimates of $p_{\eta k}(z)$ may not be as accurate. In Fig. 5 of Sect. 5 we present results of the estimations for different values of the Lipschitz constants to highlight this point. When the Lipschitz constants are not known, they can be set to larger values initially. It is worth noting that for a given number of data samples, with smaller values of the Lipschitz constants the results would be smoother. Knowing the Lipschitz continuity provides extra information to the combiner about the performance of the experts as a function of the context. This information limits the possibilities for the probabilities of false alarms and detection of the experts to the set of functions satisfying the Lipschitz continuity constraints. However, knowing this information is not critical in the detection of the labels or the estimation of the parameters. When this information is not available at the combiner, it can be set to a large number relative to the domain of context, $\mathcal{Z}$. In this way, no constraint will be applied to the EM algorithm. However to achieve a performance similar to the case that the Lipschitz constants are known, more data samples will be required.

Similar to other estimation methods which do not have access to any prior information about the probabilities of false alarm and detection of classifiers, there is always an ambiguity between two global maximums in the optimization function. Assume that P, Φ, and their entries $p_{\eta k}(z)$ and $\phi_\eta(i)$ are the actual parameters for the model. Then the probability of observing Y given P, Φ, Z is obtained by,

$$p(Y|P,\Phi;Z) = p\left(Y|\check{P},\check{\Phi};Z\right) \tag{28}$$

where $\check{P}$ is a matrix defined as $\check{p}_{\eta k}(z) \triangleq p_{(1-\eta)k}(z)$ and $\check{\Phi}$ is a matrix defined as $\check{\phi}_\eta(i) \triangleq \phi_{(1-\eta)}(i)$. Therefore $\{P,\Phi\}$ and $\{\check{P},\check{\Phi}\}$ are both candidates for the final estimation. To resolve this ambiguity, it is assumed that for the majority of classifiers, the probability of detection is greater than the probability of false alarm, i.e., the majority of classifiers have a performance above the chance line.

3.2 Combiner's decisions

In the previous section, we evaluated the estimates of probabilities of false alarm and detection for all the experts as well as the prior probabilities of the true labels $\tilde{\Phi}$.

After estimating the parameters, one can use the estimated probabilities of false alarm and detection to detect the current labels for $1 \le i \le I$ as well as the upcoming labels for $i > I$. Based on the maximum likelihood (ML) rule and for all i, the detection of a label can be performed as follows:

$$\tilde{y}(i) = \begin{cases} 1, & \prod_{k=1}^{K} p_{1k}^{\hat{y}_k(i)}(Z(i))\,(1-p_{1k}(Z(i)))^{1-\hat{y}_k(i)} \\ & \ge \prod_{k=1}^{K} p_{0k}^{\hat{y}_k(i)}(Z(i))\,(1-p_{0k}(Z(i)))^{1-\hat{y}_k(i)} \\ 0, & \text{otherwise.} \end{cases} \tag{29}$$

It is well known that the sequence of estimates obtained from the EM algorithm converge to a fixed point (Bishop 2006). Moreover, at this fixed point the derivative of the likelihood function is zero. This point may be the global maximum of the likelihood function in which case the fixed point obtained from EM is in fact the maximum likelihood estimate. However, it is also possible that the fixed point is a local maximum or a saddle point of the likelihood function. In this section we assume that the EM algorithm does indeed converge to the maximum likelihood estimate.[10] Therefore, for $1 \leq i \leq I$, we can represent the estimated parameters as the ones which maximize the following,

$$\begin{aligned}\log p\left(Y|\Theta, \mathbf{Z}\right) &= \log \sum_{\Delta} p\left(Y, \Delta \mid \Theta, \mathbf{Z}\right)\\ &= \log \sum_{\delta_0(1)} \cdots \sum_{\delta_0(I)} \prod_{i=1}^{I} \prod_{k=1}^{K} \prod_{\eta=0}^{1} \left(p_{\eta k}^{\hat{y}_k(i)}(Z(i)) \times \left(1 - p_{\eta k}(Z(i))\right)^{1-\hat{y}_k(i)} \phi_{\eta}^{\frac{1}{K}}(i)\right)^{\delta_\eta(i)}\\ &= \sum_{i=1}^{I} \log \sum_{\delta_0(i)} \prod_{k=1}^{K} \prod_{\eta=0}^{1} \left(p_{\eta k}^{\hat{y}_k(i)}(Z(i)) \times \left(1 - p_{\eta k}(Z(i))\right)^{1-\hat{y}_k(i)} \phi_{\eta}^{\frac{1}{K}}(i)\right)^{\delta_\eta(i)} \end{aligned} \tag{30}$$

where as before, if $\delta_0(i) = 1$ then $\delta_1(i) = 0$, and if $\delta_0(i) = 0$ then $\delta_1(i) = 1$. Now (30) can be written as follows.

$$\begin{aligned}\log p(Y|\Theta, \mathbf{Z}) = \sum_{i=1}^{I} \log \Bigg(&\phi_0(i) \prod_{k=1}^{K} p_{0k}^{\hat{y}_k(i)}(Z(i))\,(1 - p_{0k}(Z(i)))^{1-\hat{y}_k(i)}\\ &+ \phi_1(i) \prod_{k=1}^{K} p_{1k}^{\hat{y}_k(i)}(Z(i))\,(1 - p_{1k}(Z(i)))^{1-\hat{y}_k(i)}\Bigg)\end{aligned} \tag{31}$$

To maximize (31) with respect to

$$\left\{\phi_\eta(i), \eta = 0, 1, i = 1, 2, \ldots, I\right\},$$

it is sufficient to maximize each term in the summation. Let

$$\mathcal{A}(i) \triangleq \prod_{k=1}^{K} p_{0k}^{\hat{y}_k(i)}(Z(i))\,(1 - p_{0k}(Z(i)))^{1-\hat{y}_k(i)} \tag{32}$$

$$\text{and } \mathcal{B}(i) \triangleq \prod_{k=1}^{K} p_{1k}^{\hat{y}_k(i)}(Z(i))\,(1 - p_{1k}(Z(i)))^{1-\hat{y}_k(i)}. \tag{33}$$

Then the argument in the summation can be written as

$$\mathcal{U} = \log\left(\mathcal{A}(i)\phi_0(i) + \mathcal{B}(i)\phi_1(i)\right) \tag{34}$$

Maximizing $\mathcal{U}$ with respect to $\phi_0(i)$ and $\phi_1(i)$ with the constraint that $\phi_0(i) + \phi_1(i) = 1$, we see that the solution is either $\phi_0(i) = 1 - \phi_1(i) = 1$ (if $\mathcal{A}(i) > \mathcal{B}(i)$) or $\phi_0(i) = 1 - \phi_1(i) = 0$ (if $\mathcal{A}(i) \leq \mathcal{B}(i)$). This means that for $1 \leq i \leq I$, to detect the true labels, one can simply use the following rule,

$$\tilde{y}(i) = \begin{cases} 1, & \tilde{\phi}_1(i) \geq \tilde{\phi}_0(i) \\ 0, & \text{otherwise.} \end{cases} \tag{35}$$

We denote the final detected labels by $\tilde{\mathbf{y}} = [\tilde{y}(1),\ \tilde{y}(2),\ \ldots,\ \tilde{y}(I)]$. The entire procedure of estimating the parameter set and making decisions is summarized in Algorithm 1.

[10] All our numerical results verify this to be the case.

Algorithm 1 Estimation of the parameter set and combiner's decisions

Input: The local decisions of K experts from 1 to I, Y and the corresponding contexts, $\mathbf{Z}$
Output: The estimation of the probabilities of false alarm and detection for all of the experts, $\tilde{P}$, and the made decisions, $\tilde{\mathbf{y}}$

```
Assume an initial estimation for $\Theta^{\text{new}}$[11]
while Stopping criterion is not satisfied do
    $p_{\eta k}^{\text{old}}(Z(i)) \leftarrow p_{\eta k}^{\text{new}}(Z(i))$
    $\phi_{\eta}^{\text{old}}(i) \leftarrow \phi_{\eta}^{\text{new}}(i)$
    Find $\gamma(\eta, i)$ using (11)
    Find $\phi_{\eta}^{\text{new}}(i)$ and $p_{\eta k}^{\text{new}}(Z(i))$ using (14) and (20)
end while
For all $z \in \mathcal{Z}$, interpolate the values of $p_{\eta k}^{\text{new}}(z(1+l))$ using (25)–(27)
$\tilde{\Theta} \leftarrow \Theta^{\text{new}}$
Make decisions using (35)
```

4 Feature selection

In this section, we extend the proposed approach in Sect. 3 in order to extract the *importance* of each individual feature of a data set in the decisions of the individual experts as well as the combiner's decisions. Suppose that the received data is described by N_F features or attributes. We denote the ℓth feature by $x^{\ell} \in \mathcal{X}^{\ell}$ where $\mathcal{X}^{\ell}$ denotes the set of values that feature x^{ℓ} may assume. Hereafter, $x^{\ell} = x$ implies that the value of the ℓth feature, x^{ℓ}, is x. The system model is the same as that in Fig. 1. Each expert sends its decisions to the combiner and the combiner implements the proposed approach described in Sect. 3 once for each feature, where a feature in this section is the same as a context in Sect. 3.

We assume that the ensemble system is constructed from a variety of experts and, with the proposed approach, a very accurate detection performance can be achieved. In other words, we assume that the combiner's decisions are the same as the true labels. This assumption allows us to analyze each feature independently of the others in terms of the information between the feature and the actual label.

In the following, we find the amount of information that the local decision of the kth expert provides about the final decision of the combiner when $x^{\ell} = x$. Let $\hat{y}_k$ denote the local decision of the kth expert which is used by the combiner to make the final decision $\tilde{y}$. The mutual information between $\tilde{y}$ and $\hat{y}_k$ given that the ℓth feature takes the value x is given by

$$\begin{aligned} I\left(\tilde{y}; \hat{y}_k \mid x^{\ell} = x\right) &= \sum_{\eta=0}^{1} \sum_{j=0}^{1} p\left(\tilde{y} = \eta, \hat{y}_k = j \mid x^{\ell} = x\right) \\ &\quad \times \log \frac{p\left(\hat{y}_k = \eta, \tilde{y} = j \mid x^{\ell} = x\right)}{p\left(\hat{y}_k = j \mid x\right) p\left(\tilde{y} = \eta \mid x^{\ell} = x\right)} \\ &= \sum_{\eta=0}^{1} \sum_{j=0}^{1} p\left(\hat{y}_k = j \mid \tilde{y} = \eta, x^{\ell} = x\right) p\left(\tilde{y} = \eta \mid x^{\ell} = x\right) \\ &\quad \times \log \frac{p\left(\hat{y}_k = j \mid \tilde{y} = \eta, x^{\ell} = x\right)}{p\left(\hat{y}_k = j \mid x^{\ell} = x\right)} \end{aligned} \tag{36}$$

[11] Although we have used the symbol I for the number of indexes as well as the mutual information, we believe from the context it should be clear which notation is in use.

If we assume that the combiner is error-free and therefore its decisions are the same as the true labels, we can write

$$\begin{aligned} I\left(\tilde{y}; \hat{y}_k \mid x^{\ell} = x\right) = & \left(1 - p_{0k}\left(x;\ell\right)\right)\left(1 - \pi_{\tilde{y}}\left(x;\ell\right)\right) \log \frac{1 - p_{0k}\left(x;\ell\right)}{1 - \pi_{\hat{y}_k}\left(x;\ell\right)} && (37) \\ & + p_{0k}\left(x;\ell\right)\left(1 - \pi_{\tilde{y}}\left(x;\ell\right)\right) \log \frac{p_{0k}\left(x;\ell\right)}{\pi_{\hat{y}_k}\left(x;\ell\right)} \\ & + \left(1 - p_{1k}\left(x;\ell\right)\right) \pi_{\tilde{y}}\left(x;\ell\right) \log \frac{1 - p_{1k}\left(x;\ell\right)}{1 - \pi_{\hat{y}_k}\left(x;\ell\right)} \\ & + p_{1k}\left(x;\ell\right) \pi_{\tilde{y}}\left(x;\ell\right) \log \frac{p_{1k}\left(x;\ell\right)}{\pi_{\hat{y}_k}\left(x;\ell\right)} && (38) \end{aligned}$$

where we have assumed that

$$p\left(\hat{y}_k = j \mid \tilde{y} = \eta, x^{\ell} = x\right) = p\left(\hat{y}_k = j \mid y = \eta, x^{\ell} = x\right) \tag{39}$$

and where $p_{0k}(x;\ell)$ and $p_{1k}(x;\ell)$ are the false alarm and detection probabilities of expert k when feature ℓ is in effect and the value of this feature is x, i.e.,

$$p_{1k}(x;\ell) \triangleq p\left(\hat{y}_k = 1 \mid y = 1, x^{\ell} = x\right) \tag{40}$$

$$p_{0k}(x;\ell) \triangleq p\left(\hat{y}_k = 1 \mid y = 0, x^{\ell} = x\right) \tag{41}$$

Moreover, $\pi_{\hat{y}_k}(x;\ell) \triangleq p(\hat{y}_k = 1 \mid x^{\ell} = x)$ is the prior probability of the local decision of the kth expert given that the ℓth feature $x^{\ell} = x$, and $\pi_{\tilde{y}}(x;\ell) \triangleq p(\tilde{y} = 1 \mid x^{\ell} = x)$ is the prior probability of the final decision given that the ℓth feature $x^{\ell} = x$. For an error-free combiner which makes correct decisions in (almost) all the cases by fusing the local decisions of experts, the final decision can be considered to be independent of the feature; i.e., $\pi_{\tilde{y}}(x;\ell) \approx p(\tilde{y} = 1)$. We can estimate these two prior probabilities from our proposed algorithm (for the ℓth feature $x^{\ell} = x$) according to

$$\pi_{\tilde{y}}(x;\ell) = \frac{1}{I} \sum_{i=1}^{I} \tilde{\phi}_1(i) \tag{42}$$

$$\begin{aligned} \pi_{\hat{y}_k}(x;\ell) &= \sum_{j=0}^{1} p\left(\hat{y}_k = 1, \tilde{y} = j \mid x^{\ell} = x\right) \\ &= \sum_{j=0}^{1} p\left(\hat{y}_k = 1 \mid \tilde{y} = j, x^{\ell} = x\right) p\left(\tilde{y} = j \mid x^{\ell} = x\right) \\ &= \tilde{p}_{0k}(x)\left(1 - \pi_{\tilde{y}}(x;\ell)\right) + \tilde{p}_{1k}\left(x;\ell\right) \pi_{\tilde{y}}\left(x;\ell\right) && (43) \end{aligned}$$

As a criterion for feature selection, we define the importance of the ℓth feature for the kth expert, denoted by $\mathrm{imp}(\ell;k)$, as the correlation coefficient between the information that the kth expert provides for the true label when we use the ℓth feature as context. This is given by

$$\mathrm{imp}(\ell;k) \triangleq \left| \frac{\int_{\mathcal{X}^{\ell}} \left(I(\tilde{y}; \hat{y}_k \mid x^{\ell} = x) - \mu^{I}_{k\ell}\right)\left(x - \mu^{F}_{\ell}\right) dx}{\sigma^{I}_{k\ell}\ \sigma^{F}_{k}} \right| \tag{44}$$

where

$$\begin{aligned}
\mu_{k\ell}^{I} &= \frac{1}{\int_{\mathcal{X}^\ell} dx} \int_{\mathcal{X}^\ell} I\left(\tilde{y}; \hat{y}_k \mid x^\ell = x\right) dx \\
\mu_{\ell}^{F} &= \frac{1}{\int_{\mathcal{X}^\ell} dx} \int_{\mathcal{X}^\ell} x dx \\
\left(\sigma_{k\ell}^{I}\right)^2 &= \frac{1}{\int_{\mathcal{X}^\ell} dx} \int_{\mathcal{X}^\ell} \left(I\left(\tilde{y}; \hat{y}_k \mid x^\ell = x\right) - \mu_{k\ell}^{I}\right)^2 dx \\
\left(\sigma_{\ell}^{F}\right)^2 &= \frac{1}{\int_{\mathcal{X}^\ell} dx} \int_{\mathcal{X}^\ell} \left(x - \mu_{\ell}^{F}\right)^2 dx
\end{aligned} \tag{45}$$

If the importance of the ℓth feature for the kth expert, $\mathrm{imp}(\ell; k)$ is small, it implies that the variations of this feature will not have a significant effect on the performance of the expert, in turn indicating that this feature is not very important for that expert. On the other hand, larger values of $\mathrm{imp}(\ell; k)$ imply that this feature provides more information about the decisions of that expert. This point is further illustrated in the numerical results in Sect. 5. Finally, we define the importance of the ℓth feature as,

$$\mathrm{imp}(\ell) = \max_k \ \mathrm{imp}(\ell; k) \tag{46}$$

For a system with a large number of various experts, $\mathrm{imp}(\ell)$ shows the maximum information that can be provided by that feature from any expert among all possible experts. Therefore, if one has the ℓth feature available and has to relay on a single expert, the maximum information that can be provided by this feature about the actual label is $\mathrm{imp}(\ell)$.

The unsupervised feature selection method described above can be extended to more than one feature per expert by letting the context be formed from the set of features of interest.

5 Numerical results

In this section, we first use a system with up to 8 experts to evaluate the performance of the proposed unsupervised ensemble approach. The probabilities of false alarm and detection of these 8 experts as a function of the context z are shown in Table 1. These probabilities are selected in a way that they can represent a variety of behaviors. Many of the experts have varying accuracies with different context values, and for many values of the context the false alarm and detection probabilities of the experts are close to 0.5, i.e., these experts are not very effective in detecting the true labels. Finally the $\mathcal{L}_1$ norm is used as the distance measure, i.e., $d_{\mathcal{Z}}(z_1, z_2) = \|z_1 - z_2\|_1$.

The majority rule is the most widely used fusion rule for ensemble learning since the earliest studies of the subject (Blum 1995; Breiman 1996; Canzian et al. 2013; Fan et al. 1999; Freund and Schapire 1997; Hadavandi et al. 2015; Herbster and Warmuth 1998; Littlestone and Warmuth 1994; Schapire 1990; Stahl et al. 2015; Wang et al. 2003, 2015). In majority rule, the combiner's final decision is made by taking a vote among all the experts at each instant. That is, the final decision is the one that the majority of the experts agree on. This decision and the corresponding context is recorded for I consecutive instances. The probability of detection of each expert for a given context is estimated as the fraction of instances where the decision from the expert and the final decision from the majority rule agree and are both one for that context. Similarly, the false alarm probability of an expert for a given context is estimated as the fraction of instances that the expert's decision was one, whereas the final

Table 1 The probabilities of false alarm and detection of the classifiers

	$p_{0k}(z)$	c_{0k}	$p_{1k}(z)$	c_{1k}
Expert 1	$-2z^2 + 2z$	2.0	$.5 + .5\,\lvert\sin(2\pi z)\rvert$	3.1
Expert 2	$2(z - .5)^2$	2.0	.9	0.1
Expert 3	$.5\,\lvert\sin(2\pi z)\rvert$	3.1	$1 - 2(z - .5)^2$	2.0
Expert 4	.1	0.1	$.5 + 2(z - .5)^2$	2.0
Expert 5	$.5z$	0.5	$.75 + 2(z - .5)^3$	1.5
Expert 6	$.25 + 2(z - .5)^3$	1.5	$.75 - 2(z - .5)^3$	1.5
Expert 7	$.5(1 - z)$	0.5	$.75 + .5(z - .5)$	0.5
Expert 8	$.25 + 2(z - .5)^3$	1.5	$.5(2 - z)$	0.5

decision from the majority rule was a zero for that context. In the following we also compare the performance of the proposed method with that of the majority rule.

We initialize the EM algorithm[12] with all the probabilities of false alarm equal to 0.2, all the probabilities of detection equal to 0.8, and $\phi_1(i) = 0.6$ for $i = 1, 2, \ldots, I$. We used 100 samples randomly selected from $\mathcal{Z} = \{0, .05, .1, \ldots, 1\}$. To show the convergence speed of the proposed approach, we use a system with 4 experts; namely Experts 1–4 from Table 1. The estimated probabilities of false alarm and detection for all the experts are shown in Fig. 3 for 1, 2 and 5 iterations of the EM algorithm. For this experiment, we choose the Lipschitz constants from Table 1. It can be seen that the difference between the estimations after the 2nd and the 5th iterations are very small, indicating the fast convergence of the proposed approach. The results of using the majority rule are also shown in Fig. 3. This figure demonstrates that the performance of the proposed combiner (in estimating the false alarm and detection probabilities of the experts) is good even for a small set of experts. On the other hand for the majority rule, the final estimated probabilities are very jagged, and in all of the cases, the proposed approach significantly outperforms the majority rule. In the rest of this section, we set the number of iterations to 5.

We define the probability of error as

$$p_e = p\left(\tilde{y}(i) \neq y(i)\right) \tag{47}$$

In Fig. 4, we show the probability of error versus the number of instances I for the ensemble system with Experts 1–4 from Table 1. The results are obtained from averaging 10^4 independent trials. It can be seen that the proposed method significantly outperforms the majority rule. In Fig. 4 we also show the standard deviation for the probability of error. It can be seen that as expected, the standard deviation decreases with I and that the standard deviation of the proposed method is smaller than that of the majority rule.

The Lipschitz constant c depends on the data at hand. Figure 5 shows the effect of c on the final estimations. It can be seen that, a system with too small a constant ($c = 0.3$ here), cannot estimate the actual probabilities, even though it is very smooth. On the other hand, a constant which is too large ($c = 2.7$ here) causes jagged estimations.

[12] The initial values for the parameter set can be arbitrary. However as it is normally expected the probabilities of detection are greater than 0.5 and the probabilities of false alarm are less than 0.5. Therefore, in Sect. 5 we set all the initial probabilities of detection to 0.8 and all the probabilities of false alarm to 0.2. The labels prior probabilities, $\phi_\eta(i)$, can be any number in the interval $[0, 1]$. We have initialized $\phi_1(i)$ to 0.6 for $i = 1, 2, \ldots, I$. We also used the number of iterations as the stopping criterion and observed that good results can be obtained with 5 or fewer iterations.

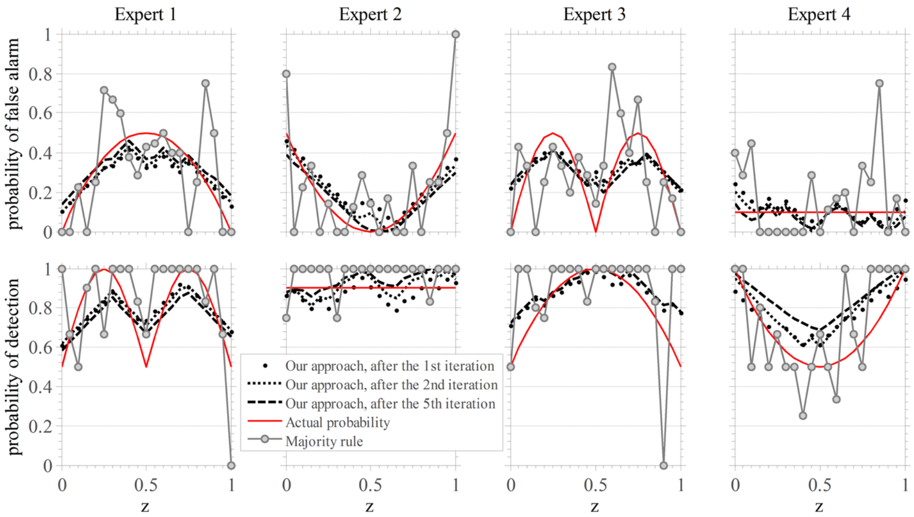

Fig. 3 Comparison of the estimations of the probabilities of false alarm and detection by using our method and majority rule versus context for $K = 4$ different experts (Experts 1–4 from Table 1), $I = 100$ after *different number* of iterations of the EM algorithm

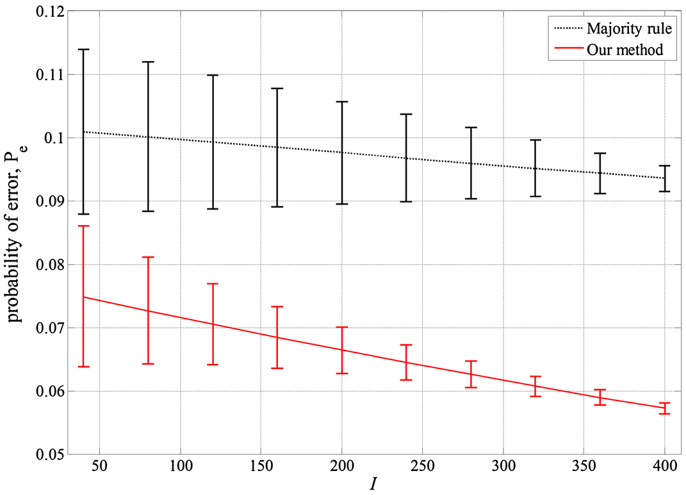

Fig. 4 The probability of error for the fusion center versus I for the ensemble system with Experts 1–4 from Table 1

Next, we evaluate the performance of the proposed approach when the Lipschitz constants are not available at the combiner. We use Expert 1–4 from Table 1 and since $\mathcal{Z} = \{0, .05, .1, ..., 1\}$ and a probability is always in $[0,\ 1]$, we assume the Lipschitz constants for all the experts are $1/0.05 = 20$ which is the maximum possible value for the Lipchitz constants in this set. We run the simulation for $I = 64, 256, 1024, 4096$. The results with 5 iterations of the EM algorithm are shown in Fig. 6. It can be seen that in the absence of any knowledge about the Lipschitz constants, as I increases, the proposed approach still converges to the actual parameters. However, in this case more data samples are required.

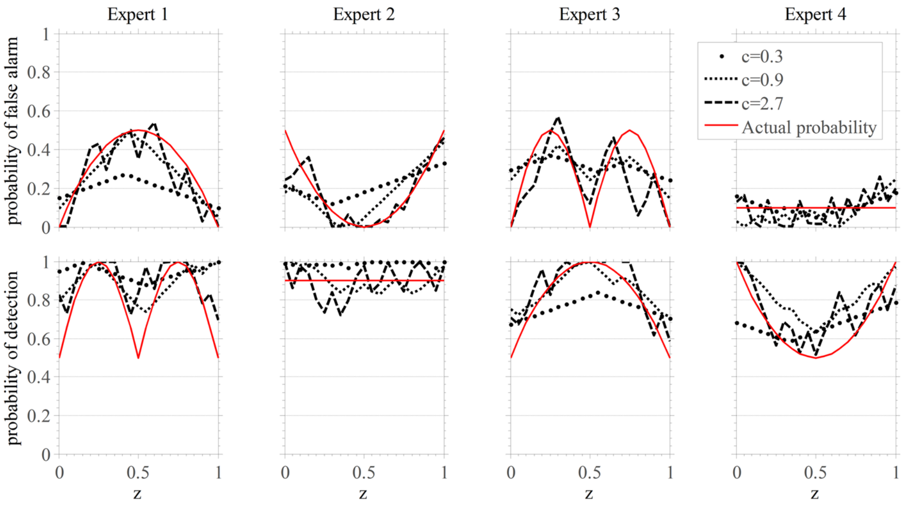

Fig. 5 Estimations of the probabilities of false alarm and detection versus context for $K = 4$ different experts (Expert 1–4 from Table 1), $I = 100$ after 5 iterations of EM algorithm using our approach, and for $c = 0.3, 0.9, 2.7$

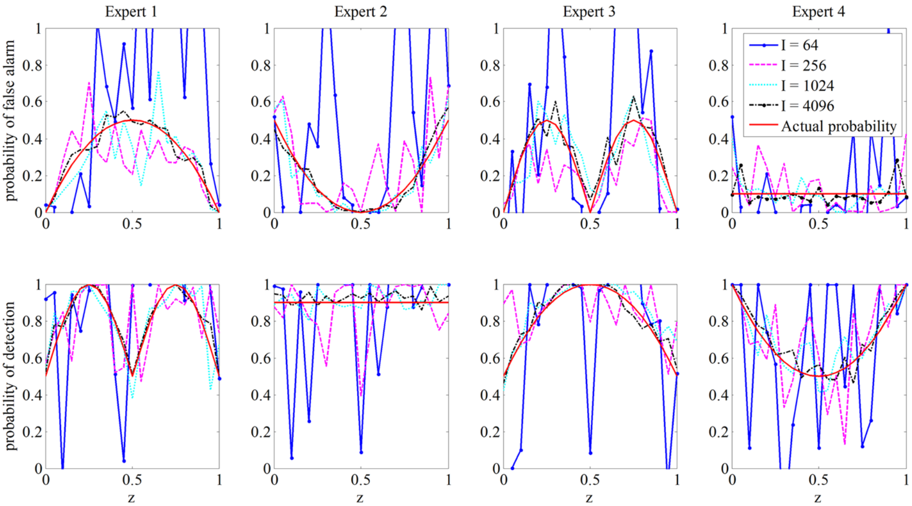

Fig. 6 Results from the proposed method for Experts 1–4 from Table 1. The Lipschitz constants for all the experts are set to 20 which is the maximum possible value for this setting. The simulation is run for $I = 64, 256, 1024, 4096$

To compare the performance of the proposed approach with the majority rule, we define a *reliability* measure for the performance for the combiner, denoted by D_P and defined below.

$$D_P \triangleq \frac{1}{2K} \sum_{k=1}^{K} \sum_{\eta=0}^{1} \frac{\int_z \left| p_{\eta k}(z) - \hat{p}_{\eta k}(z) \right| dz}{\int_z p_{\eta k}(z) dz} \tag{48}$$

In Fig. 7, we evaluate the performance of the proposed approach and the majority rule as a function of I for different number of experts, where for $K = k$, Experts $1, 2, \ldots, k$ are

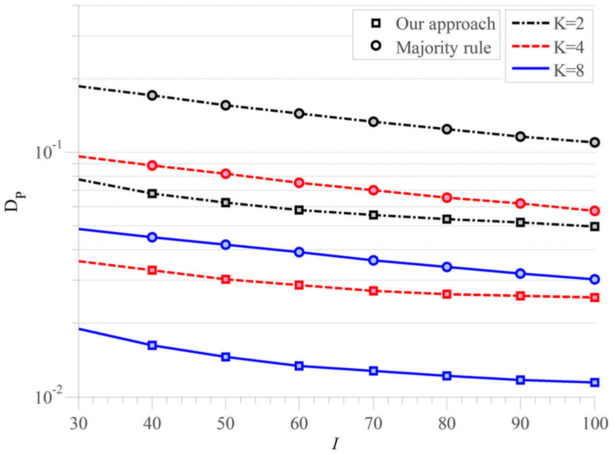

Fig. 7 Reliability, D_P versus I for $K = 2, 4, 8$ experts

used. The reliability of the combiner is shown in Fig. 7 versus I for $K = 2, 4, 8$ classes. The results are obtained from averaging 10^4 independent trials. As shown, the performance of the combiner improves with the number of experts and I and the proposed approach outperforms the majority rule in all cases.

In order to evaluate the performance of the proposed approach for real data, we used the Wisconsin breast cancer data set (Murphy and Aha 1994). The goal is to classify each data point as benign or malignant. Each data point in the data set has 9 different features: (1) clump thickness, (2) uniformity of cell size, (3) uniformity of cell shape, (4) marginal adhesion, (5) single epithelial, (6) bare nucleoli, (7) bland chromatin, (8) normal nucleoli, and (9) mitoses. All the features are in the interval [1, 10]. We used DecisionStump (one-level decision tree), KNN (k-nearest neighbor classifier), k-Star (instance classifier using entropy as distance), LogitBoost + ZeroR (ZeroR classifier uses mode), Multilayer Perceptron, and NaiveBayes (naive Bayes classifier) as experts.[13] We trained each expert with 16 samples randomly selected from the data set but with equal representation of each class. We used 241 samples from each class to perform the tests. Each of these features is considered as context separately, but due to space limitations in Figs. 8 and 11, we show the performance for clump thickness, uniformity of cell size, bland chromatin, normal nucleoli, and mitoses. We implemented our approach for each of the contexts and for $c = 0.05$, and the final results in terms of probabilities of false alarm and detection versus context are shown in Fig. 8. As shown, the NaiveBayse has the worst performance. When the context is set to be clump thickness, the performance of k-Star deteriorates with increasing z, in the sense that the probability of false alarm increases while the probability of detection does not change. Therefore, if one wishes to use one of the experts, it can be suggested that for larger values of clump thickness, it is better to use Multilayer Perceptron than k-Star.

The result of using majority rule to estimate the performance of different classifiers is shown in Fig. 9. It can be seen that compare to Fig. 8, the estimated probabilities from the majority rule are very jagged.

To evaluate the ability of the fusion rule in making the right decision about benign or malignant samples, we compare the performance of our approach against the supervised and

[13] We used machine learning classifiers from Weka. Detailed description of each classifier can be found in Witten et al. (2011).

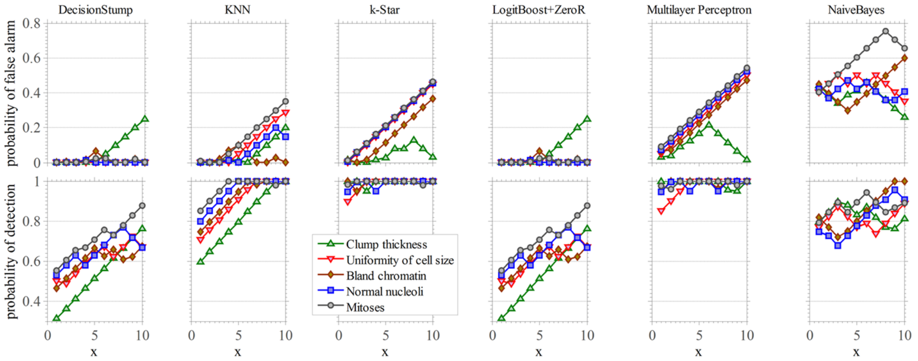

Fig. 8 The proposed approach is used in order to evaluate the performance of different experts as stated on the top of the sub-figures. Wisconsin breast cancer data set (Murphy and Aha 1994), is used in order to classify the samples into benign and malignant. Each sample point has 9 features: (1) clump thickness, (2) uniformity of cell size, (3) uniformity of cell shape, (4) marginal adhersion, (5) single epithelial, (6) bare nucleoli, (7) bland chromatin, (8) normal nucleoli, and (9) mitoses. All the features values are in the interval [1, 10]. Before running the approach each classifier has been trained with a small subset of data. The sub-figures show the probabilities of false alarm and detection versus x, where x represents: clump thickness, uniformity of cell size, bland chromatin, normal nucleoli, and mitoses (due to space limitations only these six contexts are selected to be shown here). In this experiment, we set $c = 0.05$. We should point out that the experts use all the 9 features all the time; however, at each experiment only one of the features is available as a context for the combiner and the rest are unknown

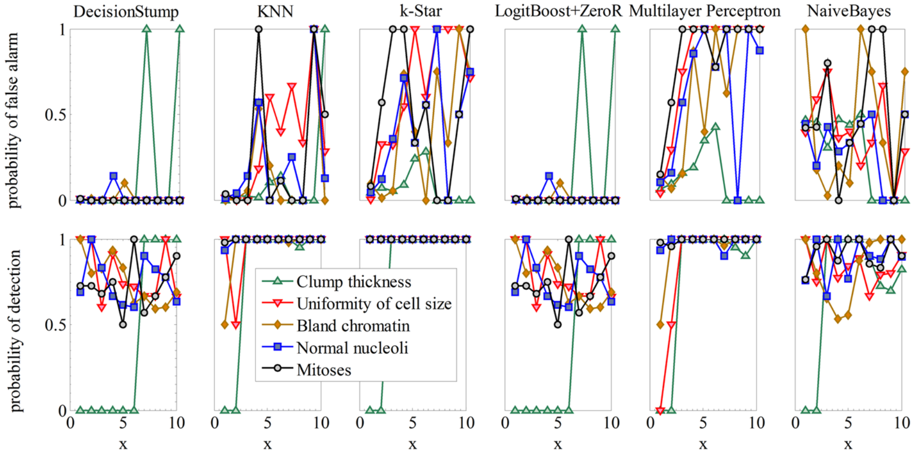

Fig. 9 Majority rule is used in order to estimate the performance of different classifiers identified at the *top* of each subfigure as a function of the feature, x

unsupervised versions of the method of tracking the best expert (MTBE) (Herbster and Warmuth 1998), adaptive Perceptron weighted majority rule (APMR) (Canzian et al. 2013), and the supervised optimal fusion rule (SOFR) (Chair and Varshney 1986), in term of probability of error, p_e. In MTBE, at each instance, the decision of each expert is compared against the actual label in the supervised version (or the pool of the decisions in the unsupervised version). A coefficient is associated with each expert and determines the weight of the expert in the pooling. This weight is updated at each instance using a nonlinear function based on how close the latest decision from the expert was to the actual label in the supervised

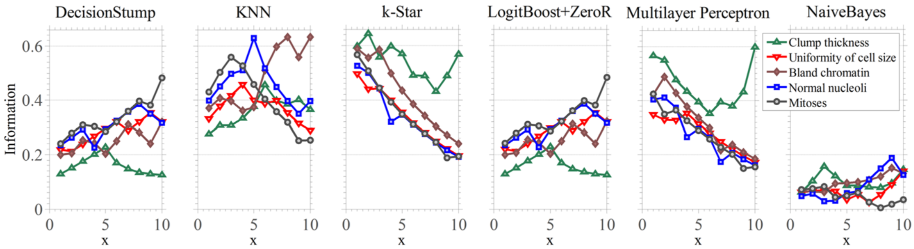

Fig. 10 Comparison of our approach with the method of tracking the best expert (MTBE), adaptive Perceptron weighted mejority rule (APMR), and supervised optimal fusion rule (SOFR) in terms of probability of correct decision versus I. We used the Wisconsin breast cancer data set and applied the same 6 experts of DecisionStump, KNN, k-Star, LogitBoost + ZeroR, Multilayer Perceptron, and NaiveBayes to obtain the results

version (or to the pool of decisions in unsupervised version). We use MTBE for the comparison as one the seminal and widely used effective online learning approaches. APMR is similar to MTBE, where the experts weights are updated using a linear function of the previous decision and the pool or actual decision. However in supervised APMR, the update for an expert weight happens only when the combiner decision is different from the actual label. We compare our approach against APMR as one of the newer suggested approaches for online learning based on Perceptron. SOFR is a supervised method in the sense that the combiner knows the performance of all the experts in the system. It uses ML rule to fuse the decision at each instant. The error rate of SOFR can be considered as a lower bound to any (supervised or unsupervised) method to compare against. In Fig. 9, the results of the comparison of these approaches and our proposed approach are shown. It can be seen that the proposed approach works better than MTBE and APMR methods including the supervised MTBE. APMR and MTBE do not fuse the data optimally. Moreover, in its modeling, APMR does not “reward” or “punish” the experts who make decision similar to or different from the combiner even when the combiner correctly detects the true label. Another fundamental problem with the unsupervised MTBE and APMR is in their modeling methods. Suppose an expert can correctly detect the event (detection probability of one) but has poor performance when the true label is 0 (large false alarm probability). Since the model used in MTBE and APMR only considers the correct detection, it can not properly characterize this expert.

Clearly, if a supervised method optimally fuses the data, its performance would be better than our proposed method as is the case with SOFR. In our proposed method, the combiner (using the EM algorithm) estimates the parameters of the system to make the final decision regarding the labels. The performance of our parameter estimation improves with the number of samples. Therefore, as I increases, the performance of our method in estimating the probabilities of false alarm and detection approaches that of the optimal ML estimator. On the other hand, as the estimates approach the actual values, the performance of our fusion rule approaches that of the ML detection rule. The proposed method is inferior to SOFR due to the fact that SOFR implements ML detection rule based on the perfect knowledge of the probabilities of false alarm and detection.

After estimating the probabilities of false alarm and detection and the prior probabilities, we evaluate the amount of information provided from each expert and for each feature. Figure 11 shows the information $I(\tilde{y}; \hat{y}_k \mid z^{\ell} = z)$, between the local decision of each of the 6 experts and the final combiner’s decision given the value of the feature, and calculated

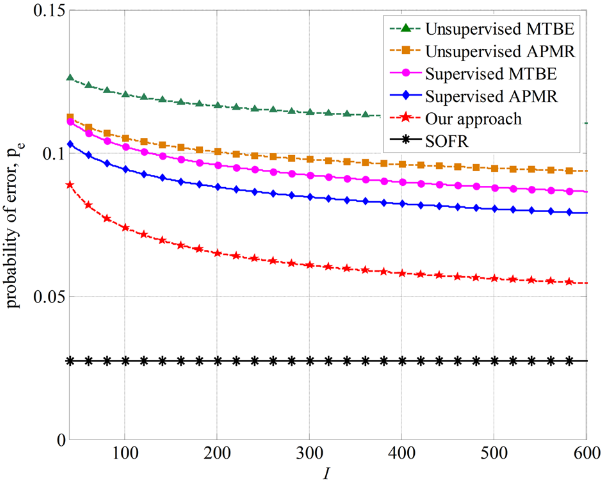

Fig. 11 The information provided from the local decision of each expert about the final decision given the value of feature in use, $I(\tilde{y}; \hat{y}_k \mid z^\ell = z)$. The results are obtained from the final estimation results of Wisconsin breast cancer data set that are shown and explained in Fig. 8

using (36). For example, when Mitoses is considered to be the context, and its value is observed to be 10 then the outputs of DecisionStump and LogitBoost + ZeroR provide the best information about the combiner's final decision.

From these results on the information provided from the local decisions of each expert given the value of the feature, we calculate the importance of each feature for every expert, $\text{imp}(\ell; k)$. The final results are shown in Table 2.[14] As an example, for KNN, the best feature is bland chromatin while for k-Star, the best feature is uniformity of cell size. The last column shows the importance of each feature, $\text{imp}(\ell)$. It can be seen that the most important feature is uniformity of cell size and the least important one is clump thickness.

According to the value of $\text{imp}(\ell)$ for different $\ell = 1, 2, \ldots, L$, we rank the features from 1 to 9, where the smaller number indicates the better feature. The result is shown in Table 3. Table 3 also shows the ranking results from mutual information quotient (MIQ) and mutual information difference (MID) (Ding and Peng 2003; Peng et al. 2005). MIQ and MID are two well-known and effective supervised methods. With the Lipchitz constants $c = 0.01, 0.05, 0.5$ the results of our unsupervised method, in which we do not observe (use) the true labels, are close to the supervised methods MIQ and MID in which the true labels are used. However, if we set the Lipschitz constant to be too small, for example, $c = 0.005$, then the final result would be less accurate. From this result and those in Fig. 6 it is evident that in the absence of any knowledge about the Lipschitz constant, it should be set to a large value. Here we also evaluate the performance of the ensemble learning when the experts in the system are not well trained, which degrades the overall performance of the combiner. With 16 training samples the probability of error for the combiner is equal to 0.057 with $I = 600$ (see Fig. 10). To reduce the performance of the combiner we trained the experts with as few as 6 and 4 samples. This increased the probability of error to 0.137 and 0.331, respectively, with $I = 600$ samples. The final result of the feature rankings are shown in Table 3. From the result in the table, it can be concluded that as the experts become less reliable in making correct decisions, which increases the overall error probability of the combiner, the ranking of the features becomes less accurate. It is worth noting that when the error rate of the system

[14] While in the first six columns we show only two digits after the decimal point, in the last column four digits are shown to more clearly distinguish the results.

Table 2 The importance of each feature for every expert, imp(ℓ; k)

	DecisionStump	KNN	k-Star	LogitBoost + ZeroR	Multilayer Perceptron	NaiveBayes	Importance, imp(k)
Clump thickness	.32	.72	.67	.32	.25	.12	*0.7158*
Uniformity of cell size	.92	.48	.995	.92	.95	.4	*0.9950*
Uniformity of cell shape	.8	.24	.994	.8	.96	.4	0.9940
Marginal adhersion	.23	.65	.994	.24	.97	.86	0.9942
Single epithelial	.88	.38	.99	.88	.93	.69	0.9897
Bare nucleoli	.37	.82	.99	.37	.98	.86	0.9904
Bland chromatin	.73	.88	.98	.73	.96	.94	0.9834
Normal nucleoli	.78	.34	.97	.78	.95	.83	0.9682
Mitoses	.92	.86	.99	.92	.99	.82	0.9880

The last column shows the importance of each feature, imp(ℓ). The italicized show the best and worst features

Table 3 Comparison of our proposed (unsupervised) feature selection method with mutual information quotient (MIQ) and mutual information difference (MID), Ding and Peng (2003) and Peng et al. (2005)

	Our approach						MIQ	MID
	$c = 0.005$	0.01	0.05			0.5		
	$p_e = 0.05$	0.05	0.331	0.137	0.05	0.05		
Clump thickness	9	9	9	9	9	9	9	9
Uni. of cell size	1	1	4	1	1	1	1	1
Uni. of cell shape	2	3	1	3	3	3	3	3
Marginal adhersion	6	2	3	2	2	2	4	2
Single epithelial	3	7	5	5	5	5	6	5
Bare nucleoli	5	4	7	4	4	4	2	4
Bland chromatin	8	8	6	8	7	7	5	6
Normal nucleoli	7	6	8	6	8	8	7	8
Mitoses	4	5	2	7	6	6	8	7

is low, then the final decision of the combiner can be considered as the true label which is independent of context (which is the same as feature in this example); more precisely, in this case Eq. (42) holds. In other words, the proposed feature selection method in Sect. 4 works well and, as shown in Table 3, the final feature ranking will be accurate.

6 Conclusion

In this paper, we provided an approach to estimate the accuracies of experts in ensemble-based decision systems and to make final decision based on the local decisions of experts. Moreover, since in many applications (especially medicine) the true label may be unknown, the proposed approach is unsupervised. Our approach does not assume any prior information about how the experts process the data to issue their decisions or their accuracies. The results show the efficiency and accuracy of the proposed approach in decision making and learning systems as well as for extracting the importance of each data feature. The proposed method has many applications, including clinical decision support systems, surveillance systems, transportation systems etc. The methods introduced in this paper can be extended in numerous directions. Subsequently, we only describe a few. First, in the current system, the experts are fixed and not adapting their expertise (rules) over time. Future work will investigate the case in which experts change and adapt their expertise over time and the impact this has on the ensemble operation and its performance. Second, in certain applications such as predictions from social media, from financial or from transportation data, the experts may significantly differ in terms of the quantity and quality of the data available to them. In such settings, it may be important to adapt the operation of the proposed ensemble scheme to take such variations into consideration. Finally, while the current experts are computer systems (machine learning algorithms), future systems may consider local experts to be a mixture of humans and computer systems. Understanding in which settings and for which applications it is beneficial to adopt ensembles of both human and computerized experts represents yet another interesting direction of future research.

Acknowledgments M. van der Schaar acknowledges the support of NSF grant ECCS 1462245.

References

Anstreicher, K. M. (1999). Linear programming in o ([n3/ln n] l) operations. *SIAM Journal on Optimization*, *9*(4), 803–812.

Bishop, C. M. (2006). *Pattern recognition and machine learning (information science and statistics)*. Secaucus, NJ: Springer.

Blum, A. (1995). Empirical support for winnow and weighted-majority algorithms: Results on a calendar scheduling domain. In *Proceedings of 12th International Conference on Machine learning* (pp. 64–72). San Francisco, CA: Morgan Kaufmann.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. New York, NY: Cambridge University Press.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140.

Brown, B., Chui, M., & Manyika, J. (2011). Are you ready for the era of big data? *McKinsey Quarterly*, *4*, 24–35.

Canzian, L., Zhang, Y., & van der Schaar, M. (2013). Ensemble of distributed learners for online classification of dynamic data streams. Preprint. arXiv:1308.5281.

Chair, Z., & Varshney, P. K. (1986). Optimal data fusion in multiple sensor detection systems. *IEEE Transactions on Aerospace and Electronic Systems*, *AES–22*(1), 98–101.

Choromanska, A., & Monteleoni, C. (2012). Online clustering with experts. In *International conference on artificial intelligence and statistics* (pp. 227–235).

Craig, T., & Ludloff, M. E. (2011). *Privacy and big data*. Sebastopol: O'Reilly Media.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B, 39*(1), 1–38.

Ding, C., & Peng, H. (Aug. 2003). Minimum redundancy feature selection from microarray gene expression data. In *Proceedings of the 2003 IEEE Bioinformatics conference, 2003. CSB 2003* (pp. 523–528).

Fan, W., Stolfo, S. J., & Zhang, J. (1999). The application of AdaBoost for distributed, scalable and on-line learning. In *Proceedings of the 5th ACM SIGKDD international conference on knowledge discovery and data mining*, KDD '99 (pp. 362–366). New York, NY:ACM.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, *55*(1), 119–139.

Hadavandi, E., Shahrabi, J., & Shamshirband, S. (2015). A novel boosted-neural network ensemble for modeling multi-target regression problems. *Engineering Applications of Artificial Intelligence*, *45*, 204–219.

Herbster, M., & Warmuth, W. K. (1998). Tracking the best expert. *Machine Learning*, *32*(2), 151–178.

Herbster, M., & Warmuth, W. K. (2001). Tracking the best linear predictor. *The Journal of Machine Learning Research*, *1*, 281–309.

Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, *11*(1), 63–90.

Huang, Y. S., & Suen, C. Y. (1995). A method of combining multiple experts for the recognition of unconstrained handwritten numerals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *17*(1), 90–94.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation*, *3*, 79–87.

Kannan, S. S., & Ramaraj, N. (2010). A novel hybrid feature selection via symmetrical uncertainty ranking based local memetic search algorithm. *Knowledge-Based Systems*, *23*(6), 580–585.

Karegowda, A. G., Manjunath, A. S., & Jayaram, M. A. (2010). Comparative study of attribute selection using gain ratio and correlation based feature selection. *International Journal of Information Technology and Knowledge Management*, *2*(2), 271–277.

Kleinberg, R., Slivkins, A., & Upfal, E. (2008). Multi-armed bandits in metric spaces. In *Proceedings of the 40th annual ACM symposium on Theory of computing* (pp. 681–690). ACM.

Kuncheva, L. I. (2004). *Combining pattern classifiers: Methods and algorithms*. Hoboken: Wiley.

Kuncheva, L. I., & Whitaker, C. J. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, *51*(2), 181–207.

Lienhart, R., Liang, L., & Kuranov, E. R. (2003). A detector tree of boosted classifiers for real-time object detection and tracking. In *IEEE international conference on multimedia and systems (ICME2003)*.

Littlestone, N., & Warmuth, M. K. (1994). The weighted majority algorithm. *Information and Computation*, *108*(2), 212–261.

Monteleoni, C., & Jaakkola, T. S. (2004). Online learning of non-stationary sequences. In S. Thrun, L. K. Saul, & B. Schölkopf (Eds.), *Advances in neural information processing systems 16* (pp. 1093–1100). Cambridge: MIT Press.

Murphy, P. M., & Aha, D. W. (1994). *UCI repository of machine learning databases: Machine readable data repository*. Irvine: Univ. of California at Irvine.

Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(8), 1226–1238.

Roobaert, D., Karakoulas, G., & Chawla, N. V. (2006). Information gain, correlation and support vector machines. In I. Guyon, S. Gunn, M. Nikravesh, & L. A. Zadeh (Eds.), *Feature extraction* (pp. 463–470). Berlin, Heidelberg: Springer-Verlag.

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, *5*(2), 197–227.

Segaran, T., & Hammerbacher, J. (2009). *Beautiful data: The stories behind elegant data solutions*. Sebastopol: O'Reilly Media, Inc.

Stahl, F., May, D., Mills, H., Bramer, M., & Gaber, M. M. (2015). A scalable expressive ensemble learning using random prism: A mapreduce approach. In A. Hameurlain, J. Küng , R. Wagner, S. Sakr, L. Wang, & A. Zomaya (Eds.), *Transactions on large-scale data-and knowledge-centered systems* (pp. 90–107). Springer.

Tekin, C., & van der Schaar, M. (2013). Distributed online big data classification using context information. Preprint. arXiv:1307.0781.

Tseng, V. S., Lee, C.-H., & Chen, J. C.-Y. (2008). An integrated data mining system for patient monitoring with applications on asthma care. In *21st IEEE international symposium on computer-based medical systems, 2008. CBMS '08* (pp. 290–292).

Wang, H., Fan, W., Yu, P. S., & Han, J. (2003). Mining concept-drifting data streams using ensemble classifiers. In *Proceedings of the 9th ACM SIGKDD international conference on knowledge discovery and data mining*, KDD '03 (pp. 226–235). New York, NY: ACM.

Wang, Y., Li, H., Wang, H., Zhou, B., & Zhang, Y. (2015). Multi-window based ensemble learning for classification of imbalanced streaming data. In J. Wang, W. Cellary, D. Wang, H. Wang, S-C Chen, T. Li, & Y. Zhang (Eds.), *Web information systems engineering—WISE 2015* (pp. 78–92). Switzerland: Springer International Publishing.

Webb, A. R., & Copsey, K. D. (2011). *Statistical pattern recognition*. Hoboken: Wiley.

Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data mining: Practical machine learning tools and techniques: Practical machine learning tools and techniques*. The Morgan Kaufmann series in data management systems. Elsevier Science.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, *5*(2), 241–259.

Zhang, D. T. Y., Sow, D., & van der Schaar, M. (2013) A fast online learning algorithm for distributed mining of bigdata. In *The big data analytics workshop at SIGMETRICS*, vol. 2013.

Zheng, H., Kulkarni, S. R., & Poor, H. V. (2011). Attribute-distributed learning: Models, limits, and algorithms. *IEEE Transactions on Signal Processing*, *59*(1), 386–398.