Context-based Unsupervised Data Fusion for Decision Making

Erfan Soltanmohammadi ERFAN@MARVELL.COM
Marvell Semiconductor, Inc., Santa Clara, CA, USA.

Mort Naraghi-Pour NARAGHI@LSU.EDU
Louisiana State University, Baton Rouge, LA, USA.

Mihaela van der Schaar MIHAELA@EE.UCLA.EDU
University of California, Los Angeles, CA, USA.

Abstract

Big Data received from sources such as social media, in-stream monitoring systems, networks, and markets is often mined for discovering patterns, detecting anomalies, and making decisions or predictions. In distributed learning and real-time processing of Big Data, ensemble-based systems in which a fusion center (FC) is used to combine the local decisions of several classifiers, have shown to be superior to single expert systems. However, optimal design of the FC requires knowledge of the accuracy of the individual classifiers which, in many cases, is not available. Moreover, in many applications supervised training of the FC is not feasible since the true labels of the data set are not available. In this paper, we propose an unsupervised joint estimation-detection scheme to estimate the accuracies of the local classifiers as functions of data context and to fuse the local decisions of the classifiers. Numerical results show the dramatic improvement of the proposed method as compared with the state of the art approaches.

1. Introduction

Ensemble-based approaches have proven to be more accurate than single-classifier systems for many applications involving decision making, prediction, classification, or detection (Kuncheva & Whitaker, 2003). Furthermore, for problems with high computational complexity, ensemble-based approaches allow for distributed processing which results in load sharing among the individual classifiers. Due to their high computational complexity, Big Data applications including data mining, decision making, and prediction demand parallel processing for which ensemble learning is well-suited (Zhang et al., 2013; Tekin & van der Schaar, 2013).

An ensemble system is comprised of a set of (possibly heterogeneous[1]) classifiers and a combining rule for fusing the classifiers' outputs. Individual classifiers may be trained with different data sets and by judiciously combining their outputs we can achieve a more accurate decision; a set of linear (or nonlinear) classifiers may be used to span the data space to a complicated nonlinear boundary.

The fusion center (FC) which combines the local decisions of the classifiers plays the key role in the performance of the overall system. Several different fusion rules have been proposed in the literature. The majority rule may be employed when no information on the performance of the classifiers is available (Kuncheva, 2004). On the other hand, weighted majority rule can be used when the performances of individual classifiers are known a priori. Another approach is to construct a look-up table during the test and validation procedure including the output patterns of the classifiers and their corresponding labels. This approach, known as Behavior Knowledge Space (BKS), actually estimates the densities of the classifier outputs and requires large training and validation data sets (Huang & Suen, 1995). For some ensemble systems the classifiers and the FC are trained together using a joint procedure such as stacked generalization or mixture of classifiers (Wolpert, 1992; Jacobs et al., 1991).

Optimal fusion of local decisions requires the a priori knowledge of the accuracy of the classifiers which, in many applications, may not be available. For example, the data may have an extremely large dimension which makes it im-

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37. Copyright 2015 by the author(s).

[1] Here heterogeneity of classifiers implies that they have different error rates in classifying the data (Webb & Copsey, 2011).

precise to evaluate the accuracy of the classifiers based on the training and validation sets; or the data stream may be time-varying for which accurate evaluation of the classifiers' performance is impractical.

In many applications the data stream is received along with its own context. The context may be a small side information such as a description of the way the data is acquired, (Tekin & van der Schaar, 2013), or it may be a small dimensional portion of the actual high dimensional data representing one of its features or attributes. Since the accuracies of the classifiers vary with the data context, optimal fusion rule requires knowledge of the accuracies of the classifiers for every arriving context resulting in prohibitively high costs in processing, communication and storage requirements.

In this paper we assume that no prior information regarding the classifiers' performance is available. The details on the working of each classifier, and how they receive their data is also unknown. Each classifier may work with a different part of the Big Data, the preprocessed data, or even different correlated data from distributed multiple sources. We propose an unsupervised method based on the Expectation-Maximization (EM) algorithm, (Dempster et al., 1977), for evaluating the accuracies of the classifiers as functions of context as well as fusing the decisions of individual classifiers. To this end, we introduce a model for estimation of the classifiers' accuracies in terms of probabilities of false alarm and detection. As such the proposed approach allows for maximum likelihood estimation of the classifier parameters based on unlabeled data. This model is different from other typical models in which the probability of correct decision is used to evaluate the performance of classifiers, (Tekin & van der Schaar, 2013; Canzian & van der Schaar, 2014). Our approach is also different from that in (Platanios et al., 2014) where the accuracies of classifiers are estimated for unlabeled data. The authors assume that several (at least three) classifiers operate on the same data set and the classifiers make independent errors. By calculating the agreement rates of the classifiers, the authors are able to estimate the error-rate of each classifier. This method does not work if only a single classifier operates on each data set.

2. Problem Formulation and Notations

We consider an ensemble learning system with K classifiers each classifying an input data stream characterized by its context. Every classifier makes a local decision which it delivers to the FC for the final decision. Since multiple-choice decision making 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. Let the portion of data available for the kth classifier be denoted by $s_k(t) \in \mathcal{S}_k$, and let $X(t) \in \mathcal{X}$ be the context of the received data where t is the integer-valued time index [2]. The context, which may be a vector in general, may represent a side information about the data or it may be a subset of the features (attributes) of the data. For instance, in the case of image labeling, the context may be the camera resolution. The set $\mathcal{X}$ is assumed to be a (subset of a) metric space with the metric $d_{\mathcal{X}}(x_1, x_2)$ that represents the distance between x_1 and x_2. Let $y(t) \in \mathcal{Y} \triangleq \{0, 1\}$ denote the true label at time t. In the proposed approach, the true label $y(t)$ is not available for training. Moreover, we are not concerned about how the classifiers classify the data. However, the accuracy of each classifier is estimated as a function of the context $X(t)$.

Let $\mathbf{X}(t_0) \triangleq [X(t_0), X(t_0+1), \cdots, X(t_0+T-1)]$ and $\mathbf{y}(t_0) \triangleq [y(t_0), y(t_0+1), \cdots, y(t_0+T-1)]$ denote the observed vector of contexts and the unobserved vector of true labels, respectively, for a duration T starting at t_0. As mentioned previously, $\mathbf{y}(t_0)$ is not available and its detection is also a part of the proposed approach. Note that in this and subsequent sections, t is in the range t_0 to $t_0 + T - 1$, k goes from 1 to K, and i goes from 0 to 1. We define the *label matrix*, by $\Delta(t_0) = [\delta_i(t)]_{2\times T}$, where column t of Δ corresponds to the true label $y(t)$, and at each time t, one of the elements in column t is 1 and the other is 0. If $\delta_0(t) = 0$, then $\delta_1(t) = 1$, indicating that at time t we have $y(t) = 1$; similarly, if $\delta_0(t) = 1$, then $\delta_1(t) = 0$, indicating that at time t we have $y(t) = 0$.

Let $\hat{y}_k(t)$ be the local decision of the kth classifier at time t and let $\hat{\mathbf{y}}(t) = [\hat{y}_1(t)\ \hat{y}_2(t)\ \ldots\ \hat{y}_K(t)]^\dagger$ denote the vector of K local decisions at time t, where † represents the transpose operation. Finally, let $\hat{Y}(t_0) = [\hat{y}_k(t)]_{K\times T}$ denote the collection of local decisions of all classifiers for duration T. The FC receives the decisions of all the classifiers, $\hat{Y}(t_0)$, (as well as the context $\mathbf{X}(t_0)$) and needs to fuse them to get an estimate of the true labels. However, for judicial fusing of the received decisions, the FC must estimate the accuracy of each classifier.

To model the accuracy of the classifiers, we associate a probability of detection and a probability of false alarm with each classifier.

Since the performance of a classifier depends on the context of the data it receives, these probabilities are assumed to be functions of the context. For a fixed context, however, a classifier has fixed probabilities of detection and false alarm. Therefore, for context x and for classifier k, we define the probability of detection, denoted by $p_{1k}(x)$,

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

and the probability of false alarm, denoted by $p_{0k}(x)$ as

$$p_{ik}(x) \triangleq p(\hat{y}_k(t) = 1 \mid \delta_i(t) = 1; x), \;\; i = 0, 1 \qquad (1)$$

We assume that the probability $p_{ik}(x)$ is Lipschitz continuous with Lipschitz constants c_{ik}, i.e.,

$$|p_{ik}(x_1) - p_{ik}(x_2)| \; \leq \; c_{ik} \, d_{\mathcal{X}} \left(x_1, x_2\right) \qquad (2)$$

This assumption which imposes a constraint on how fast a classifier's accuracy can change with context is clearly valid in most practical situations, (Kleinberg et al., 2008; Tekin & van der Schaar, 2013). For instance, in the case of image labeling, where the context may be the camera resolution, it is not expected that the accuracy of a classifier can change sharply with a small change in the resolution of the images. We arrange these probabilities for all the classifiers into a matrix $P(x) \triangleq [p_{ik}(x)]$, $i = 0, 1$, $k = 1, 2, \ldots, K$. Note that the FC does not know $P(x)$ and one of the goals of our proposed method is to estimate it. In addition, in the formulation above and the proposed solution, the context variable x may be vector-valued. For example, if several features are considered as part of the context, then x will be a vector. In this case, the metric $d_X(x_1, x_2)$ may be chosen to be an L_μ norm for some $\mu \geq 1$.

In order to facilitate the detection of the true labels we assign probabilities $\phi_0(t)$ and $\phi_1(t)$ to label $y(t)$ and arrange them in a matrix $\Phi(t_0) = [\phi_i(t)]_{2\times T}$, where $\phi_i(t) = p(\delta_i(t) = 1)$ and $\phi_0(t) + \phi_1(t) = 1$. We should point out that the probabilities $\phi_i(t)$ do not represent a prior probability of the true labels. They are introduced in order to convert the problem of detection of the true labels into the problem of estimation of the $\phi_i(t)$'s which is then solved using the EM algorithm. Also, please note that neither $\Delta(t_0)$ nor $\Phi(t_0)$ are available to the FC. They are assumed to be unknown parameters which are evaluated in the proposed method in order to estimate $P(x)$ and to detect $\hat{Y}(t_0)$. To summarize, the two-tuple, $\Theta = \{P(x), \Phi(t_0)\}$ is defined as the unknown *parameter set* which the FC tries to estimate based on the local decisions of the classifier, $\hat{Y}(t_0)$, and context of the data, $\mathbf{X}(t_0)$. After estimating the parameter set Θ, the FC detects the true labels $\mathbf{y}(t_0)$. In the next section, we propose an approach based on the EM algorithm for the FC to achieve these goals.

Remark: *One may ask whether, instead of the detection and false alarm probabilities of the classifiers, their accuracies (i.e., the error probabilities) can be estimated. The problem is that for the case of unsupervised learning being considered here, we do not know how to solve the problem in terms of the error probabilities of the classifiers. Moreover, it is clear that given the detection and false alarm probabilities of individual classifiers, we can implement the maximum likelihood (ML) fusion rule. However, given the accuracies, we cannot formulate an ML fusion rule.*

3. Estimation of the Classifiers' Accuracies and Decision Making

In this section, given the local decisions, $\hat{Y}(t_0)$, and the observed vector of contexts, $\mathbf{X}(t_0)$, we first develop an estimation method for Θ. We then use the estimated $\Phi(t_0)$ to detect the true labels $\mathbf{y}(t_0)$.

3.1. Estimation Procedure

The maximum likelihood estimate of Θ given $\hat{Y}(t_0)$ and $\mathbf{X}(t_0)$ is given by

$$\hat{\Theta} = \arg\max_{\Theta} \sum_{\Delta} p(\hat{Y}(t_0), \Delta(t_0) \mid \Theta, \mathbf{X}(t_0)) \qquad (3)$$

By considering $\Delta(t_0)$ as a latent variable, the mixture model in (3) can be iteratively solved using the EM algorithm. First, we evaluate $p(\hat{Y}(t_0), \Delta(t_0) | \Theta, \mathbf{X}(t_0))$ from

$$p(\hat{Y}(t_0), \Delta(t_0) | \Theta, \mathbf{X}(t_0)) = \prod_t \prod_k \prod_i \qquad (4)$$
$$\left[p_{ik}^{\hat{y}_k(t)}(X(t)) \left(1 - p_{ik}(X(t))\right)^{1-\hat{y}_k(t)} \phi_i^{\frac{1}{K}}(t)\right]^{\delta_i(t)}$$

The log-likelihood function, $\log p(\hat{Y}(t_0), \Delta(t_0) \mid \Theta, \mathbf{X}(t_0))$, is obtained as

$$L(\Theta; \hat{Y}(t_0), \Delta(t_0), \mathbf{X}(t_0))$$
$$= \sum_k \sum_t \sum_i \delta_i(t) \Big[\hat{y}_k(t) \log p_{ik}(X(t))$$
$$+ (1 - \hat{y}_k(t)) \log\left(1 - p_{ik}(X(t))\right) + \frac{1}{K} \log \phi_i(t)\Big] \qquad (5)$$

The two steps of EM algorithm 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 $p(\Delta(t_0) \mid \hat{Y}(t_0); \Theta^{\text{old}})$ of the latent variable $\Delta(t_0)$, where Θ^{old} is the previous estimate for Θ. That is,

$$Q(\Theta; \Theta^{\text{old}}) \qquad (6)$$
$$= E_{\Delta(t_0)|\hat{Y}(t_0);\Theta^{\text{old}}} \left[L(\Theta; \hat{Y}(t_0), \Delta(t_0), \mathbf{X}(t_0))\right]$$
$$= \sum_k \sum_t \sum_i \gamma(i, t) \Big[\hat{y}_k(t) \log p_{ik}(X(t))$$
$$+ (1 - \hat{y}_k(t)) \log\left(1 - p_{ik}(X(t))\right) + \frac{1}{K} \log \phi_i(t)\Big]$$

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

$$\gamma(i, t) = E_{H|Y;\Theta^{\text{old}}}\left[\delta_{it}\right] = p(\delta_{it} = 1 \mid Y; \Theta^{\text{old}}, X(t))$$
$$= p(\delta_{it} = 1 \mid \hat{\mathbf{y}}(t); \Theta^{\text{old}}, X(t)) = \qquad (7)$$
$$\frac{\phi_i^{\text{old}}(t) \prod_k \left(p_{ik}^{\text{old}}(X(t))\right)^{y_k(t)} \left(1 - p_{ik}^{\text{old}}(X(t))\right)^{1-y_k(t)}}{\sum\limits_{j=0}^{1} \phi_j^{\text{old}}(t) \prod_k \left(p_{jk}^{\text{old}}(X(t))\right)^{y_k(t)} \left(1 - p_{jk}^{\text{old}}(X(t))\right)^{1-y_k(t)}}$$

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 $\phi_i(t)$ we must consider the constraint $\sum_{i=0}^{1}\phi_i(t) = 1$. Using the Lagrange multiplier method, we get

$$\phi_i^{\text{new}}(t) = \frac{\gamma(i,t)}{\sum_{j=0}^{1}\gamma(j,t)} = \gamma(i,t) \tag{8}$$

We would like to note that since $Q(\Theta;\Theta^{\text{old}})$ is a concave function of $\phi_i(t)$, and the constraint is linear, the above Lagrangian method results in the optimal solution for $\Phi(t_0)$.

Maximization of $Q(\Theta;\Theta^{\text{old}})$ with respect to $p_{ik}(X(t))$ is also a constraint optimization problem given by

$$p_{ik}^{\text{new}}(X(t)) = \underset{p_{ik}(X(t))}{\arg\max}\ Q(\Theta;\Theta^{\text{old}}) \tag{9}$$

subject to:

$$|p_{ik}(x_1) - p_{ik}(x_2)| \leq c_{ik} d_{\mathcal{X}}\left(x_1, x_2\right),\ \forall x_1, x_2 \in \mathcal{X}$$

$$0 \leq p_{ik}(x) \leq 1 \text{ for } i = 0,1,\ k = 1,2,\ldots,K,\ \forall x \in \mathcal{X}$$

It can be shown that $Q(\Theta;\Theta^{\text{old}})$ is a concave function of $p_{ik}(X(t))$. Therefore we can use convex optimization methods to solve (9). Towards this let

$$\varrho_{ik}(l,j) \triangleq c_{ik} d_{\mathcal{X}}\left(x(l), x(j)\right) \tag{10}$$

$$\mathbf{p}_{ik}(t_0) \triangleq \tag{11}$$

$$[p_{ik}(X(t_0)), p_{ik}(X(t_0+1)), \ldots, p_{ik}(X(t_0+T-1))]^{\dagger}$$

$$\psi(\mathbf{p}_{ik}(t_0)) \triangleq \sum_{t}\gamma(i,t)\Big(\hat{y}_k(t)\log p_{ik}(X(t)) + \tag{12}$$

$$(1-\hat{y}_k(t))\log\left(1 - p_{ik}(X(t))\right)\Big)$$

Then, to maximize $Q(\Theta;\Theta^{\text{old}})$ with respect to $p_{ik}(x)$ subject to the Lipschitz continuity constraint in (2), we can solve the constrained optimization problem given by

$$\mathbf{p}_{ik}^{\text{new}}(t_0) = \underset{\mathbf{p}_{ik}(t_0)}{\arg\max}\ \psi(\mathbf{p}_{ik}(t_0)) \tag{13}$$

subject to:

$$|p_{ik}(x(t_0+l)) - p_{ik}(x(t_0+j))| \leq \varrho_{ik}(l,j)\ \ \forall\, l, j,$$

$$0 \leq p_{ik}(X(t)) \leq 1 \text{ for } i = 0,1,\ k = 1,2,\ldots,K,$$

$$t = t_0, t_0+1, \ldots, t_0+T-1$$

We can rewrite (13) as

$$\mathbf{p}_{ik}^{\text{new}}(t_0) = \underset{\mathbf{p}_{ik}(t_0)}{\arg\max}\ \psi(\mathbf{p}_{ik}(t_0)) \tag{14}$$

$$\text{subject to: } \Lambda\mathbf{p}_{ik}(t_0) \leq \boldsymbol{\varrho}_{ik},\ \ \mathbf{0} \leq \mathbf{p}_{ik}(t_0) \leq \mathbf{1}$$

where the inequalities are component-wise, and **0** and **1** are the all-zero and the all-one column vectors of length T, respectively, and where Λ and $\boldsymbol{\varrho}_{ik}$ are defined below. Note that the objective function in (14) is concave and the constraints are linear; Therefore, (14) can be solved using the interior point methods, (Boyd & Vandenberghe, 2004).

By iterating between the expectation and the maximization steps, until a stopping criterion is satisfied[3], we find an estimation of the parameter set[4]. We denote the final estimates of the parameter set by $\tilde{\Theta}$, and the final estimates of $P(x) = [p_{ik}(x)]$ and $\Phi = [\phi_i(t)]$ by $\tilde{P(x)} = [\tilde{p}_{ik}(x)]$, and $\tilde{\Phi} = [\tilde{\phi}_i(t)]$, respectively.

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

and

$$\boldsymbol{\varrho}_{ik} \triangleq \begin{array}{c|c|}
1 & \varrho_{ik}(1,2) \\
2 & \varrho_{ik}(1,2) \\
3 & \varrho_{ik}(1,3) \\
4 & \varrho_{ik}(1,3) \\
\vdots & \vdots \\
2T-1 & \varrho_{ik}(1,T) \\
2T & \varrho_{ik}(1,T) \\ \cline{2-2}
2T+1 & \varrho_{ik}(2,3) \\
2T+2 & \varrho_{ik}(2,3) \\
2T+3 & \varrho_{ik}(2,4) \\
2T+4 & \varrho_{ik}(2,4) \\
\vdots & \vdots \\
4T-1 & \varrho_{ik}(2,T) \\
4T-2 & \varrho_{ik}(2,T) \\ \cline{2-2}
\vdots & \vdots \\ \cline{2-2}
T^2-T-1 & \varrho_{ik}((T-2),T) \\
T^2-T & \varrho_{ik}((T-1),T)
\end{array}$$

In order to evaluate $p_{ik}(x)$ for all $x \in \mathcal{X}$, we note that for

[3] A stopping criterion could be a selected number of iterations or a threshold on the percentage difference between the last two estimations.

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

any $j = t_0, t_0 + 1, \cdots, t_0 + T - 1$, $i = 0, 1$ and $k = 1, 2, \cdots, K$,

$$\tilde{p}_{ik}(x(j)) - c_{ik} d_{\mathcal{X}}\left(x, x(j)\right) \leq \tilde{p}_{ik}(x) \tag{15}$$

$$\tilde{p}_{ik}(x(j)) + c_{ik} d_{\mathcal{X}}\left(x, x(j)\right) \geq \tilde{p}_{ik}(x) \tag{16}$$

Therefore, we can interpolate the values of $p_{ik}(x(t_0 + l))$, $l = 0, 1, \cdots, T - 1$ to obtain $p_{ik}(x)$ for any $x \in \mathcal{X}$. Let

$$p1_{ik}(x) = \max_{t_0 \leq j \leq t_0 + T - 1} \left\{\tilde{p}_{ik}(x(j)) - c_{ik} d_{\mathcal{X}}\left(x, x(j)\right)\right\} \tag{17}$$

$$p2_{ik}(x) = \min_{t_0 \leq j \leq t_0 + T - 1} \left\{\tilde{p}_{ik}(x(j)) + c_{ik} d_{\mathcal{X}}\left(x, x(j)\right)\right\} \tag{18}$$

We then set[5]

$$\tilde{p}_{ik}(x) = \min\left\{p1_{ik}(x), p2_{ik}(x)\right\} \tag{19}$$

Remark: The Lipschitz constants c_{ik} affect the performance of the algorithm in estimating the parameters $p_{ik}(x)$ as functions of x. As evident from (2), (15) and (19), smaller values of c_{ik} result in a smoother estimate for $p_{ik}(x)$, while larger values of c_{ik} allow for larger variations in the estimates. Therefore the Lipschitz constants c_{ik} must be selected in accordance with the performance of the classifiers vs. the context variables. In particular, if for example the detection performance $p_{1k}(x)$ of the kth classifier is believed to be very sensitive to the context variable x, i.e., small changes in x result in large changes in $p_{1k}(x)$, then the value of c_{1k} must be chosen to be large. On the other if the detection performance of the kth classifier is not very sensitive to the context variable x, then a smaller value should be assigned to c_{1k}. That said, we would like to also point out that the Lipschitz condition in (2) is only introduced to enable the estimation of the functions $p_{ik}(x)$. If the selected parameters c_{ik} do not satisfy (2) for the true functions $p_{ik}(x)$, then our algorithm still works. However, in this case our estimates of $p_{ik}(x)$ will not be very accurate. In Fig. 2 of Section 4 we present results of the estimations for different values of the Lipschitz constants which verify this statement.

3.2. Fusion Center's Decisions

In the previous section, we evaluated the estimates of probabilities of false alarm and detection for all the classifiers as well as the prior probabilities of the true labels $\tilde{\Phi}$. To detect the true labels of the classifiers we use the maximum likelihood detection of $y(t)$ given the probabilities, $\tilde{\Phi}$, namely $\tilde{y}_{\text{EM}}(t) = \arg\max_{y(t)} p(y(t) \mid \tilde{\Theta})$ which is given by

$$\tilde{y}(t) = \begin{cases} 1, & \tilde{\phi}_1(t) > \tilde{\phi}_0(t) \\ 0, & \text{otherwise.} \end{cases} \tag{20}$$

We denote the final detected labels by $\tilde{\mathbf{y}} = [\tilde{y}(t_0),\ \tilde{y}(t_0 + 1),\ \ldots,\ \tilde{y}(t_0 + T - 1)]$. The entire procedure of estimating the parameter set and making decisions is summarized in Algorithm 1.

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

Input: The local decisions of K classifiers from t_0 to $t_0 + T - 1$, $\hat{Y}(t_0)$ and the corresponding contexts, $\mathbf{X}(t_0)$

Output: The estimation of the probabilities of false alarm and detention for all of the classifiers, $\tilde{P}$, and the made decisions, $\tilde{\mathbf{y}}$

```
Assume an initial estimation for $\Theta^{\text{new}}$
while Stopping criterion is not satisfied do
    $p_{ik}^{\text{old}}(X(t)) \leftarrow p_{ik}^{\text{new}}(X(t))$
    $\phi_i^{\text{old}}(t) \leftarrow \phi_i^{\text{new}}(t)$
    Find $\gamma(i,t)$ using (7)
    Find $\phi_i^{\text{new}}(t)$ and $p_{ik}^{\text{new}}(X(t))$ using (8) and (14)
end while
For all $x \in \mathcal{X}$, interpolate the values of $p_{ik}^{\text{new}}(x(t_0 + l))$
using (17)-(19)
$\tilde{\Theta} \leftarrow \Theta^{\text{new}}$
Make decisions using (20)
```

4. Numerical Results

In this section, first we use a system with up to 8 classifiers to evaluate the performance of the proposed approach. The probabilities of false alarm and detection of these classifiers as a function of the context x are shown in Table 1. These probabilities are selected so as to represent a variety of behaviors. In particular, the classifiers are not very accurate, and for many values of the context, their false alarm and detection probabilities are close to 0.5. The $\mathcal{L}_1$ norm is used as the distance measure, i.e., $d_{\mathcal{X}}(x_1, x_2) = \|x_1 - x_2\|_1$. The values of the Lipschitz constants are also shown in Table 1. These values are selected so as to satisfy the condition in (2).

In Fig. 1 we show the performance of the proposed method in estimating the probabilities of false alarm and detection of the individual classifiers. To show the convergence speed of the proposed approach, we use a system with 4 classifiers; namely Classifiers $1 - 4$ from Table 1. We initialize the EM algorithm with all the probabilities of false alarm equal to 0.2, all the probabilities of detection equal to 0.8, and $\phi_1(t) = 0.6$ for $t = t_0, t_0 + 1, \cdots, T + t_0 - 1$. The parameter T is chosen to be 100. The estimated probabilities of false

[5] The minimum in (19) 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 median or mean.

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

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

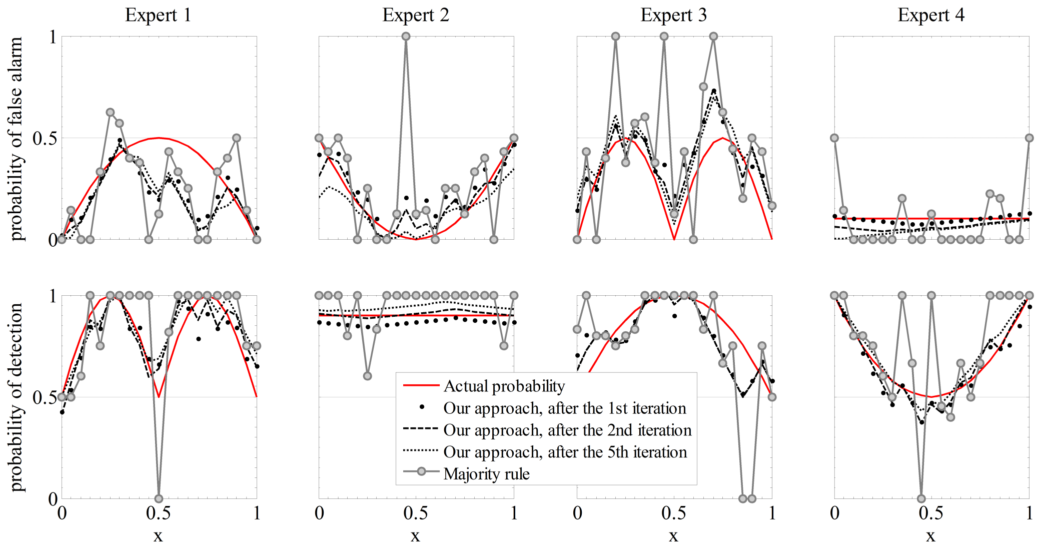

Figure 1. Comparison of the proposed method and the majority rule.

alarm and detection for the four classifiers are shown in Fig. 1 for $1, 2$ and 5 iterations of the EM algorithm. In Fig. 1 we also compare the performance of the proposed method with that of the majority rule which is the most widely used unsupervised fusion rule for ensemble learning (Breiman, 1996; Schapire, 1990; Freund & Schapire, 1997; Herbster & Warmuth, 1998; Canzian et al., 2013). 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. On the other hand for the majority rule, the final estimated probabilities are very spiky, 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.

In Fig. 2 we show the effect of the Lipschitz constants on the final estimations. Here we set $c_{ik} = c$ for all $i = 0, 1$ and $k = 1, 2, \cdots, K$. Three different values of $c = 0.2, 1.7, 3.2$ are used. It is evident that for small value of $c = 0.2$, the estimated detection and false alarm probabilities are a very smooth function of the context x. However, the estimations do not closely follow the actual functions. On the other hand, for $c = 3.2$, the estimation can better follow the rapid variations of $p_{ik}(x)$ vs. x, but in this case the estimations are somewhat spiky.

In order to quantify the improvement of the proposed method over the majority rule we define a *reliability* metric, denoted by D_P where

$$D_P \triangleq \frac{1}{2K}\sum_{k=1}^{K}\sum_{i=0}^{1}\frac{\int_x |p_{ik}(x)-\hat{p}_{ik}(x)|\,dx}{\int_x p_{ik}(x)dx} \tag{21}$$

The reliability metric in (21) measures the normalized-error in the estimation of the detection and false alarm probabilities of all the classifiers[6]. Clearly a smaller value of D_P indicates a better performance for the estimator.

In Fig. 3, we show the value of D_P vs. T for different number of classifiers, where for $K = \ell$, Classifiers $1, 2, \ldots, \ell$ are used. The values of c_{ik} are given in Table 1. As shown, the performance of our method improves with the number of classifiers and T and the proposed approach outperforms the majority rule in all cases.

In Fig. 4 we show the probability of error for the proposed method vs. the majority rule for the case presented in Fig. 1. It can be seen that the proposed method significantly outperforms the majority rule.

In order to evaluate the performance of the proposed approach for real data, we used the Wisconsin breast cancer data set (Murphy & Aha, 1994). The goal is to classify

[6] Please note that if the set of context values is discrete, then the integrals in (21) are replaced with summations.

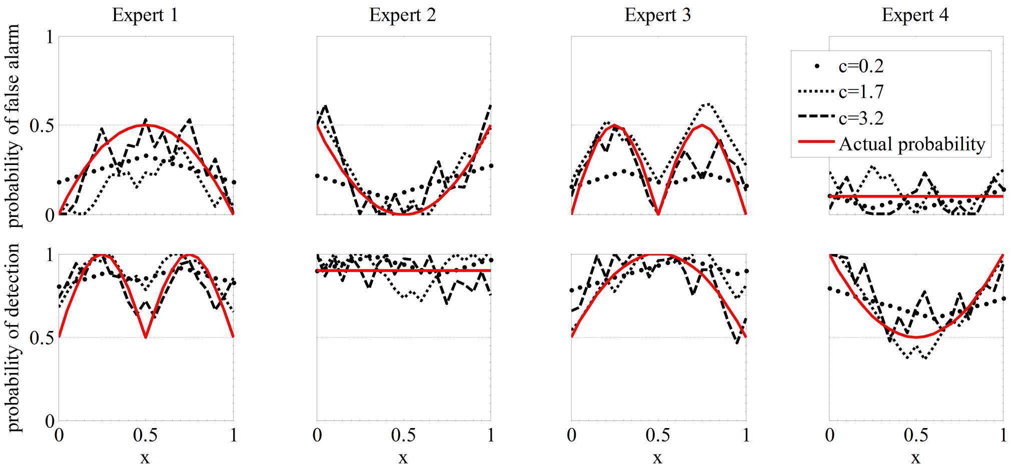

Figure 2. Estimations of the probabilities of false alarm and detection vs. context for $K = 4$ different experts (Expert1-4 from Table 1), $T = 100$ using the proposed approach for $c = 0.2, 1.7, 3.2$.

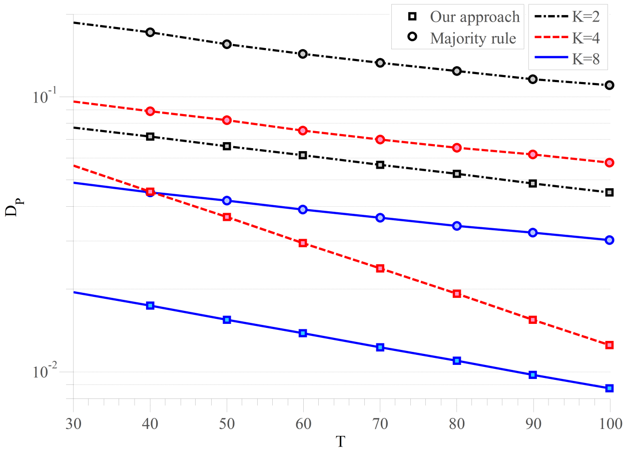

Figure 3. Reliability, D_P versus T for $K = 2, 4, 8$ classifiers. The values of cik are given in Table 1.

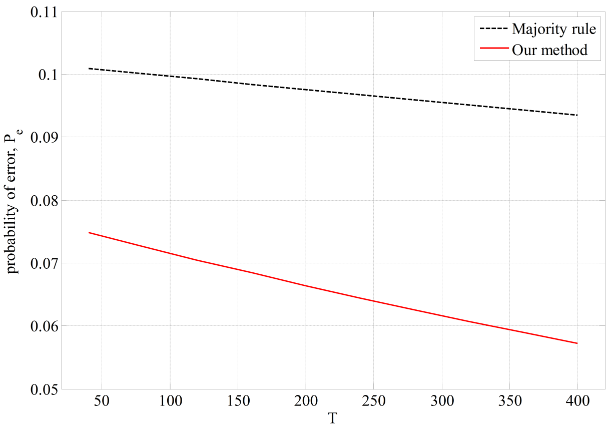

Figure 4. The probability of error for the fusion center vs. T for the case in Fig. 1.

each data point as benign or malignant.

Remark: *We should point out that this data set comes with true labels. As discussed previously, our algorithm does not require the true labels and does not utilize the labels in order to estimate the performance of the classifiers and for fusing the decisions of the individual classifiers. However, the labels are used in order to evaluate the performance of our algorithm in terms of correct decisions (see Fig. 6) as well as to compare our results with other methods. The labels are also used for training the supervised optimal fusion rule (SOFR),(Chair & Varshney, 1986), which provides a lower bound to the performance of any unsupervised technique (see Fig. 6)*

Each 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 classifiers and trained them with a subset of the data[7]. Each of these features is considered as context separately, but due to space limitations, in Fig. 5 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 where for each i and k we set the Lipschitz constants

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

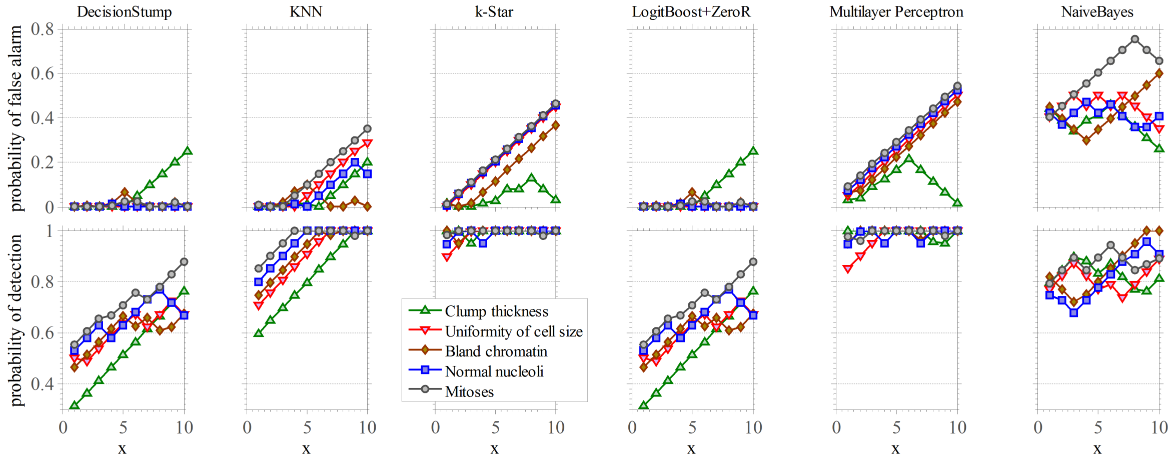

Figure 5. The proposed approach is used in order to evaluate the performance of different classifiers identified at the top of each subfigure as a function of the context x.

$c_{ik} = 0.05$[8], and the final results in terms of probabilities of false alarm and detection versus context are shown in Fig. 5. As shown, the NaiveBayes has the worst performance. When the context is set to be $x =$ clump thickness, the performance of k-Star deteriorates with increasing x, 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 classifiers, it can be suggested that for larger values of clump thickness, it is better to use Multilayer Perceptron than k-Star. Therefore the proposed method can be used in this way to determine the efficacy of the individual classifiers.

To evaluate the performance of the fusion rule in making the right decision about benign or malignant samples, we define the probability of fusion error as $p_e = p(\tilde{y}(t) \neq y(t))$. In Fig. 6, we compare the results of our unsupervised method with the supervised and unsupervised versions of the method of tracking the best classifier (MTBE), (Herbster & Warmuth, 1998), adaptive Perceptron weighted majority rule (APMR), (Canzian et al., 2013), and the SOFR, in term of p_e vs. T. The parameters for our method are the same as those in Fig. 5 with each feature used as a context with a value between 1 and 10. It can be seen that the proposed approach works better than MTBE and APMR and even the supervised MTBE. APMR and MTBE do not fuse the data optimally. Moreover, in its modeling APMR does not "reward" or "punish" the classifiers who make decisions similar to or different from the FC even when the FC correctly detects the true label. Another fundamental problem with the unsupervised MTBE and APMR is that since these methods are only concerned with correct detection, they do not properly characterize a classifier which has a very high false alarm probability.

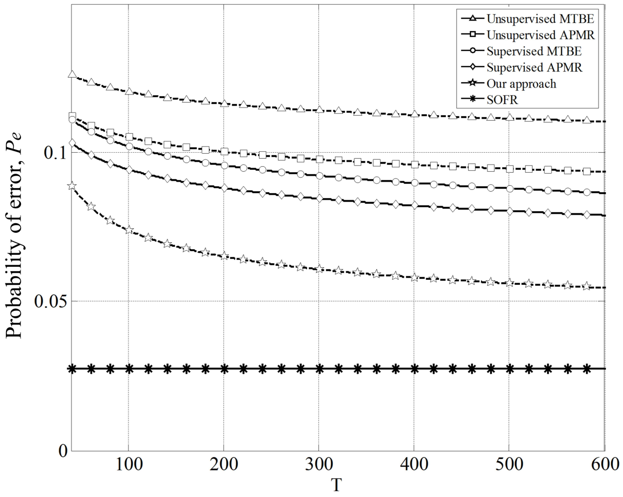

Figure 6. Comparison of our approach with the method of tracking the best classifier (MTBE), adaptive Perceptron weighted majority rule (APMR), and supervised optimal fusion rule (SOFR).

5. Conclusion

Ensemble-based systems have proven to be superior to single-expert systems for Big Data analytics. In many applications prior information about the accuracies of the individual classifiers is not available and the true label of the data is never observed. In this paper, we propose an unsupervised method to estimate the accuracies of the experts and to fuse their local decisions to obtain a final decision. The results show the superior performance of the proposed approach as compared with the state of the art approaches.

[8] For this data set and as shown n Fig. 5, the variations of false alarm and detection probabilities with respect to the context variable are very small.

References

Bishop, Christopher M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

Boyd, Stephen and Vandenberghe, Lieven. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521833787.

Breiman, Leo. Bagging predictors. *Mach. Learn.*, 24(2): 123–140, August 1996. ISSN 0885-6125. doi: 10.1023/A:1018054314350.

Canzian, Luca and van der Schaar, Mihaela. A network of cooperative learners for data-driven stream mining. In *Acoustics, Speech and Signal Processing, 2014. ICASSP 2014 Proceedings. 2014 IEEE International Conference on*, To appear 2014.

Canzian, Luca, Zhang, Yu, and van der Schaar, Mihaela. Ensemble of distributed learners for online classification of dynamic data streams. *arXiv preprint arXiv:1308.5281*, 2013.

Chair, Z. and Varshney, P.K. Optimal data fusion in multiple sensor detection systems. *Aerospace and Electronic Systems, IEEE Transactions on*, AES-22(1):98 – 101, jan. 1986. ISSN 0018-9251. doi: 10.1109/TAES.1986.310699.

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

Freund, Yoav and Schapire, Robert E. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, August 1997. ISSN 0022-0000. doi: 10.1006/jcss.1997.1504.

Herbster, Mark and Warmuth, Manfred K. Tracking the best expert. *Machine Learning*, 32(2):151–178, 1998.

Huang, Y.S. and Suen, C.Y. A method of combining multiple experts for the recognition of unconstrained handwritten numerals. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(1):90–94, 1995. ISSN 0162-8828. doi: 10.1109/34.368145.

Jacobs, Robert A, Jordan, Michael I, Nowlan, Steven J, and Hinton, Geoffrey E. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

Kleinberg, Robert, Slivkins, Aleksandrs, and Upfal, Eli. Multi-armed bandits in metric spaces. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, pp. 681–690. ACM, 2008.

Kuncheva, L.I. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley, 2004. ISBN 9780471660255.

Kuncheva, Ludmila I. and Whitaker, Christopher J. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Mach. Learn.*, 51(2):181–207, May 2003. ISSN 0885-6125. doi: 10.1023/A:1022859003006.

Lienhart, Rainer, Liang, Luhong, and Kuranov, Er. A detector tree of boosted classifiers for real-time object detection and tracking. In *IEEE Int. Conf. on Multimedia and Systems (ICME2003)*, 2003.

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

Platanios, Emmanouil Antonios, Blum, Avrim, and Mitchell, Tom M. Estimating Accuracy from Unlabeled Data. In *Conference on Uncertainty in Artificial Intelligence*, pp. 1–10, 2014.

Schapire, Robert E. The strength of weak learnability. *Mach. Learn.*, 5(2):197–227, July 1990. ISSN 0885-6125. doi: 10.1023/A:1022648800760.

Tekin, Cem and van der Schaar, Mihaela. Distributed online big data classification using context information. *arXiv preprint arXiv:1307.0781*, 2013.

Webb, A.R. and Copsey, K.D. *Statistical Pattern Recognition*. Wiley, 2011. ISBN 9781119952961.

Witten, I.H., Frank, E., and Hall, M.A. *Data Mining: Practical Machine Learning Tools and Techniques: Practical Machine Learning Tools and Techniques*. The Morgan Kaufmann Series in Data Management Systems. Elsevier Science, 2011. ISBN 9780080890364.

Wolpert, David H. Stacked generalization. *Neural Networks*, 5(2):241 – 259, 1992. ISSN 0893-6080. doi: http://dx.doi.org/10.1016/S0893-6080(05)80023-1.

Zhang, DTY, Sow, Daby, and van der Schaar, M. A fast online learning algorithm for distributed mining of bigdata. In *the Big Data Analytics workshop at SIGMETRICS*, volume 2013, 2013.