Detecting Erroneous Classifiers in Batch Distributed Inference

Yuval Shicht
Electrical and Computer Engineering
Technion
Haifa, Israel
yuval.shicht@technion.ac.il

Yuval Cassuto
Electrical and Computer Engineering
Technion
Haifa, Israel
ycassuto@ee.technion.ac.il

Abstract

Distributed inference is a promising paradigm in machine learning, but errors from participants may corrupt the inference result. To mitigate this in binary-classification tasks, we study the problem of detecting erroneous classifiers. Each classifier in a distributed ensemble provides a batch of classification outputs to a central node, and the proposed detectors aim to find the erroneous ones among them. Two types of detectors are studied: 1) blind detectors that know nothing about the statistics of the classifiers, and 2) informed-statistics detectors that know the classifiers' pair-wise agreement statistics. We develop analytical tools for evaluating the detection performance, and demonstrate the tools for ensembles following the Bernoulli-Mixture model. In addition, we provide empirical results to validate the improvement in classification accuracy on real neural-network based classifiers.

CCS Concepts

• **Computing methodologies** → **Distributed artificial intelligence**; **Ensemble methods**.

ACM Reference Format:
Yuval Shicht and Yuval Cassuto. 2025. Detecting Erroneous Classifiers in Batch Distributed Inference. In *International Workshop on Secure and Efficient Federated Learning (FL-AsiaCCS '25), August 25–29, 2025, Hanoi, Vietnam.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3709023.3737693

1 Introduction

Distributed inference is a setting in which an inference operation is performed by multiple entities residing in different locations, and it is an important building block in general distributed machine learning [7], in particular in *federated learning (FL)*. Given a data point $\boldsymbol{x}$, rather than performing inference using one (centralized) function (also called model) $F(\boldsymbol{x})$, the data point (or features thereof) is distributed across multiple nodes, each employing a partial model $h_i(\boldsymbol{x})$. The outputs of the partial models are sent to an aggregating node, which combines the partial inferences into one final inference. The advantages of distributed inference are significant; principal among them are: communication reduction and data-privacy enhancement. The challenge, however, is the possibility that some partial inferences will reach the aggregating node *with errors.*

We focus in this paper on the inference task of *binary classification.* In the studied problem, each $h_i(\cdot)$ is a binary-output partial-classification function trained using some *ensemble method* in machine learning (e.g., *bagging*, *boosting*, *decision forests*), and M such outputs are aggregated by a simple majority rule to obtain the final classification output. Some partial-classification outputs are *flipped* before reaching the aggregating node, leading to degraded classification performance. Several recent works have addressed the issue of erroneous classifiers in distributed inference: [10] proposed resource-allocation algorithms when classifiers are aggregated with heterogeneous weights, [2] developed post-training algorithms to optimize the transmission powers and aggregation weights, and [3] provided adaptive-boosting [5] training algorithms that take into account classifier errors happening at inference time. To this mitigation toolbox we add another method in this paper: *detection* of erroneous classifiers prior to aggregation.

Prior work on erroneous classifiers made classification more robust to errors, without attempting to detect errors at inference time before aggregation. Indeed, when classifying a single data point at a time, it is not clear how this can be done. However, when we classify *a batch* of data points, we can potentially use the larger amount of available information to detect a classifier that is likely erroneous. We follow this approach in this paper, focusing on detecting a single erroneous classifier out of the size-M ensemble, where it is assumed that the erroneous classifier flips outputs i.i.d. with probability ϵ. The problem of detecting erroneous classifiers from a batch is related to the well-known problem of outlier detection [1, 6], in particular outlier hypothesis testing [13] and robust learning from batches [8]. While these prior works provide good intuitions on the problem, they are not immediately applicable since erroneous classifiers may not necessarily skew the symbol distribution in the batch. Indeed, batches with balanced 0/1 labels remain balanced even with arbitrary flipping probability by the erroneous classifier. To address that problem, our scheme uses *pair-wise statistics* on classifier outputs, either inferred from the batch or known a-priori.

We propose three detectors: one is "blind": it needs no prior information on the classification statistics of the ensemble. It works by measuring the *discrepancy* between every classifier and the majority of the other classifiers. The other two detectors are from a class we call "informed statistics": they have knowledge of the pair-wise *agreement* distribution of the classifiers without errors. For the blind detector, we analyze the detection-success probability by deriving the distribution of the batch discrepancies. For the informed-statistics likelihood-based detector, we analyze the false-negative probability by deriving an approximation of the log likelihood ratio. In both cases we specialize the analysis for classifier outputs that follow a *Bernoulli-Mixture (BM)* distribution [4],

Work supported in part by the US-Israel Binational Science Foundation grant 2023627.

FL-AsiaCCS '25, Hanoi, Vietnam

ACM ISBN 979-8-4007-1420-7/25/08
https://doi.org/10.1145/3709023.3737693

whose key property is that classifier outputs are independent *given the data-point's ground-truth label.* Detection performance depends on four parameters: the *ensemble size* M, the *accuracy* of individual classifiers α, the *flipping rate* ϵ, and the *batch size* N. We validate the results using a numerical evaluation in Section 4, including on an implementation of neural-network classifier ensembles for an image classification task.

2 Problem and Detector Formulation

In a distributed-classification setting, we have an ensemble of M *base classifiers.* Each base classifier implements a binary classification function $h_i(\cdot)$, computing a value in $\{0,1\}$ for an input data point. In normal operation, these functions are aggregated into a final classification value by a majority rule:

$$H(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \sum_{i=1}^{M} h_i(\boldsymbol{x}) > M/2 \\ 0 & \text{if } \sum_{i=1}^{M} h_i(\boldsymbol{x}) < M/2 \\ u \sim \text{Bern}\left(\frac{1}{2}\right) & \text{otherwise} \end{cases} \tag{1}$$

The aggregation rule makes a deterministic 1/0 decision if there is strict majority, and in cases of ties draws the output randomly as a Bernoulli variable with parameter 1/2.

An (unknown) member of the ensemble may be *erroneous*: its outputs reach the aggregator node *flipped with probability* ϵ, independently between data points. Defined formally,

Definition 2.1. Let i' be an index of an erroneous classifier, then the value it delivers to the aggregator on data point $\boldsymbol{x}(n)$ is $h_{i'}(\boldsymbol{x}(n))+z(n)$, where $z(n)$ is a Bernoulli random variable with parameter ϵ, and $+$ denotes modulo-2 addition.

For notation simplicity, in the sequel we replace $h_i(\boldsymbol{x}(n))$ by $x_i(n)$ and by X_i when the identity of the data point is clear from the context. In order to avoid contaminating the classification with erroneous inputs, *detection* of erroneous base classifiers is desired. Classification and detection are performed in *batches* of N data points each, and the crux of detection is to use the information in a batch to detect a base classifier that is likely erroneous. The action taken following detection depends on the specific practical setting: we may conservatively decide to *completely block* the detected classifier, that is, remove the index of the suspect from the sum in (1), or alternatively take a milder measure. We pursue in this paper two types of detectors: 1) *blind detectors* that have no prior information on the classifiers, and 2) *informed-statistics detectors* that have some statistical information on the classifiers (for example, from a training dataset).

2.1 Blind Detection by Majority Discrepancy

The proposed blind detector works by comparing the classifier outputs to the majority of the remaining classifiers. A classifier that differs from the majority on many data points may indicate it being erroneous. We formalize this by the measure of *discrepancy*:

Definition 2.2. For some data point, define X_j as the value delivered by classifier j, and let $S_i \triangleq \sum_{j\neq i} X_j$. Then the **discrepancy** of classifier i on this data point is defined as

$$Y_i \triangleq \begin{cases} 1 & \text{if } X_i \neq \mathbb{1}_{S_i \geq M/2} \\ 0 & \text{otherwise.} \end{cases}$$

where $\mathbb{1}$ is the indicator function.

The discrepancy Y_i is 1 if and only if classifier i's output is different from the majority value of the *other* $M-1$ classifiers. Let $[m] \triangleq \{1,\ldots,m\}$, and now define the following detector for a single erroneous classifier.

Detector 1. *Define $y_i(n)$ to be the discrepancy of classifier i on data point n. Then, the detector output is the index*

$$\hat{i} = \underset{i\in[M]}{\text{argmax}} \sum_{n=1}^{N} y_i(n). \tag{2}$$

Detector 1 estimates the index of the erroneous classifier as the one that has the largest sum discrepancy over the N-batch.

2.2 Informed-Statistics Detection through Pair-Wise Agreement

The detectors defined in this sub-section use more refined statistics than Detector 1, in particular, they have some prior statistical knowledge on the pair-wise classifier correlations under normal (non-erroneous) operation. The following *agreement* variable is a pair-wise counterpart of the discrepancy defined in Section 2.1.

Definition 2.3. For a given data point with classifier outputs $\{X_i\}_{i=1}^{M}$, define the **agreement** variable for classifiers i,j as

$$W_{i,j} = \begin{cases} 1, & \text{if } X_i = X_j \\ 0, & \text{otherwise.} \end{cases}$$

Note that the discrepancy Y_i in Section 2.1 is a negative correlation measure (disagreement), while the agreement $W_{i,j}$ here is a positive correlation measure: equals 1 when X_i and X_j have the same value. A probabilistic model for the classifiers specifies

$$p_{W_{i,j}} \triangleq P(W_{i,j}=1) = P(X_i=0, X_j=0) + P(X_i=1, X_j=1).$$

We assume that the probabilities $p_{W_{i,j}}$, for all pairs $\{i,j\}$, $i\neq j$ are known to the detector, and $p_{W_{i,j}} \neq 0.5$. Toward defining the first informed-statistics detector, we use $w_{i,j}(n)$ to denote the empirical agreement of classifiers i,j on data point n.

Detector 2. *Given the prior classifier statistics $\{p_{W_{i,j}}\}_{\substack{i,j=1\\ i\neq j}}^{M}$ and the N-batch $\{(x_1(n),\ldots,x_M(n))\}_{n=1}^{N}$, calculate $w_{i,j}(n)$ for each $\{i,j\}_{\substack{i,j=1\\ i\neq j}}^{M}$ and $n=1,\ldots,N$. Then, the detector output is the index*

$$\hat{i} = \underset{i\in[M]}{\text{argmax}} \frac{1}{M-1} \sum_{j\neq i} \left(\frac{p_{W_{i,j}} - \frac{1}{N}\sum_{n=1}^{N} w_{i,j}(n)}{2p_{W_{i,j}} - 1} \right). \tag{3}$$

Detector 2 uses the known probabilities $p_{W_{i,j}}$, unlike Detector 1 that is blind to the classifier statistics. The following proposition justifies the use of Detector 2, at least in the case of a single erroneous classifier (proof omitted).

Proposition 2.4. *Let $f(i)$ be the function maximized by Detector 2 in* (3)*. Then for a single erroneous classifier i', the expectation of $f(i')$ equals the flipping probability ϵ, and for every $i \neq i'$ it equals $\epsilon/(M-1)$.*

By Proposition 2.4, the function of Detector 2 has a multiplicative gap of factor $M-1$ (in expectation) between the erroneous classifier and all the others, hence it is a good candidate for detection. We can further use this function as an estimate for the flipping probability of the erroneous classifier.

For a more systematic statistical treatment of the detection problem, we now pursue detectors using the tool of *likelihood tests* [12]. Given an N-batch of classifier outputs $\{(x_1(n), \dots, x_M(n))\}_{n=1}^N$ and the agreement values calculated from them: $\{w_{i,j}(n)\}_{n=1}^N$, we wish to decide which classifiers in $\{1, \dots, M\}$ are likely erroneous and which are not. Toward that, we are given the same agreement probabilities $p_{W_{i,j}}$ given to Detector 2, as well the classifier biases: $p_{X_i} \triangleq P(X_i = 1)$, for every i. We assume for non-erroneous classifiers that *independently for each data point n*, $P(x_i(n) = 1) = p_{X_i}$ and $P(w_{i,j}(n) = 1) = p_{W_{i,j}}$ (and their complements $P(x_i(n) = 0) = 1 - p_{X_i}$, $P(w_{i,j}(n) = 0) = 1 - p_{W_{i,j}}$).

Now, we want to decide whether classifier i is erroneous. Toward that, we define the i-th classifier's *marginal likelihood function* $P(X_i) \prod_{\substack{j=1 \\ j \neq i}}^M P(W_{i,j})$. Looking at this task as a binary hypothesis-testing problem, the *null hypothesis* H_0 is that classifier i is not erroneous, and we want to distinguish it from the *alternative hypothesis* H_1 in which i is erroneous. In both cases we assume that the remaining classifiers are not erroneous. In the case of H_0 the distribution of the batch's data points is known: $P(X_i) = p_{X_i}$ and $P(W_{i,j}) = p_{W_{i,j}}$. Denote $Q_i \triangleq \sum_{n=1}^N x_i(n)$ and $Q_{i,j} \triangleq \sum_{n=1}^N w_{i,j}(n)$; from the independence between data points, we get that the marginal likelihood of the N-batch for H_0 is

$$L_{\text{batch}-i|\text{null}} = \left(p_{X_i}\right)^{Q_i} \left(1 - p_{X_i}\right)^{N-Q_i} \prod_{\substack{j=1 \\ j \neq i}}^{M} \left(p_{W_{i,j}}\right)^{Q_{i,j}} \left(1 - p_{W_{i,j}}\right)^{N-Q_{i,j}} . \tag{4}$$

For the H_1 hypothesis, we need to replace $P(X_i)$ and $P(W_{i,j})$ in the marginal likelihood by different probabilities q_i and $q_{i,j}$, respectively. Since the true probabilities are unknown, we take the probabilities that maximize the likelihood of the batch. It is well known [12, Ch.14] that these equal $\hat{q}_i \triangleq Q_i/N$, and similarly $\hat{q}_{i,j} \triangleq Q_{i,j}/N$. This gives the batch maximum likelihood of H_1

$$\max L_{\text{batch}-i} = \left(\hat{q}_i\right)^{Q_i} \left(1 - \hat{q}_i\right)^{N-Q_i} \prod_{\substack{j=1 \\ j \neq i}}^{M} \left(\hat{q}_{i,j}\right)^{Q_{i,j}} \left(1 - \hat{q}_{i,j}\right)^{N-Q_{i,j}} . \tag{5}$$

We define classifier i's batch likelihood ratio (LR) as the ratio between (5) and (4)

$$LR_{\text{batch}-i} \triangleq \frac{\max L_{\text{batch}-i}}{L_{\text{batch}-i|\text{null}}} = \left(\frac{\hat{q}_i}{p_{X_i}}\right)^{Q_i} \left(\frac{1 - \hat{q}_i}{1 - p_{X_i}}\right)^{N-Q_i} \prod_{\substack{j=1 \\ j \neq i}}^{M} \left(\frac{\hat{q}_{i,j}}{p_{W_{i,j}}}\right)^{Q_{i,j}} \left(\frac{1 - \hat{q}_{i,j}}{1 - p_{W_{i,j}}}\right)^{N-Q_{i,j}} .$$

Taking the log of the batch likelihood ratio, we get

$$LLR_{\text{batch}-i} = Q_i \log\left(\frac{\hat{q}_i}{p_{X_i}}\right) + (N - Q_i) \log\left(\frac{1 - \hat{q}_i}{1 - p_{X_i}}\right) + \sum_{\substack{j=1 \\ j \neq i}}^{M} \left[Q_{i,j} \log\left(\frac{\hat{q}_{i,j}}{p_{W_{i,j}}}\right) + (N - Q_{i,j}) \log\left(\frac{1 - \hat{q}_{i,j}}{1 - p_{W_{i,j}}}\right)\right] . \tag{6}$$

Equivalently, $LLR_{\text{batch}-i} = N[D_{KL}(\hat{Q}_{X_i} \| P_{X_i}) + \sum_{\substack{j=1 \\ j \neq i}}^M D_{KL}(\hat{Q}_{W_{i,j}} \| P_{W_{i,j}})]$, where $D_{KL}(\cdot \| \cdot)$ are the *Kullback–Leibler (KL) divergences* between the binary distributions in the arguments: $\hat{Q}_{X_i}$ is Bern$(\hat{q}_i)$, P_{X_i} is Bern(p_{X_i}), $\hat{Q}_{W_{i,j}}$ is Bern$(\hat{q}_{i,j})$ and $P_{W_{i,j}}$ is Bern$(p_{W_{i,j}})$. Note that from the non-negativity of the KL divergence, $LLR_{\text{batch}-i}$ is always non-negative; the larger it is, the lower is the likelihood that the null hypothesis is true. We are now ready to specify the next detector.

Detector 3. *Given $\{Q_i\}_{i=1}^M$ and $\{Q_{i,j}\}_{\substack{i,j=1 \\ i \neq j}}^M$ calculated from the N-batch, and given the prior classifier statistics $\{p_{X_i}\}_{i=1}^M$ and $\{p_{W_{i,j}}\}_{\substack{i,j=1 \\ i \neq j}}^M$: find*

$$\hat{i} = \operatorname*{argmax}_{i \in [M]} LLR_{\text{batch}-i} . \tag{7}$$

Then declare classifier $\hat{i}$ erroneous if $LLR_{\text{batch}-\hat{i}} > \eta$, for some specified threshold $\eta(M)$.

Detector 3 first identifies the classifier $\hat{i}$ that is most "anomalous" with respect to the likelihood ratio test, and then declares it erroneous *if this ratio is above the threshold.* This thresholding step offers the advantageous choice to not declare any erroneous classifier when not needed, while the previous Detectors 1,2 by definition always declare an erroneous classifier.

3 Analysis of the Detectors

In this section we derive analytical results for the detection performance of Detectors 1 and 3. For concreteness, we focus in the analysis on classifier ensembles modeled as *Bernoulli Mixtures (BM)*, hence we first include the definitions of the BM model (BMM).

3.1 The Bernoulli Mixture Model (BMM) Classifier Distribution

The BMM is a common model for binary data [9], where in this paper we use it to model the outputs of the functions $\{h_i\}_{i=1}^M$ on a data point, which we recall to be denoted $(X_i)_{i=1}^M$ in this paper. At a high level, the BMM captures the dependence between the M classifiers when predicting the label of the data point. Let $\boldsymbol{X} = (X_1, \dots, X_M)$ be an M-variate BM random vector with parameters $\alpha \in [0, 1]$ and $\pi \in [0, 1]$. This defines the following: first *a label* is chosen to be 1 with probability π and 0 with probability $1 - \pi$. Then each X_i is drawn independently as a Bernoulli random variable with the following rule:

$$P(X_i = 1) = \begin{cases} \alpha & \text{if label is 1} \\ 1 - \alpha & \text{if label is 0} \end{cases} , \; P(X_i = 0) = \begin{cases} 1 - \alpha & \text{if label is 1} \\ \alpha & \text{if label is 0} \end{cases}$$

The label models the ground-truth classification of each data point. The shared rule introduces dependence between the variables, but *conditioned on the label* the classifier outcomes $X_1, \dots, X_M$ are independent. We say that classifier i is *correct* if X_i agrees with the label, which happens with probability α, and *incorrect* if otherwise, which happens with probability $1 - \alpha$. Hence the value of α represents the *accuracy* of the individual classifier functions. Unless otherwise noted, we fix $\pi = 0.5$ to model the case where the 1-labels and 0-labels are balanced in the dataset. The above model we consider in the paper is a special case of the general BM model where each variable can have a different parameter α_i.

3.2 Analysis of the Blind Detector under BMM

The key statistic used by Detector 1 is the discrepancy defined in Definition 2.2, and summed in (2). We analyze the detection-success probability of Detector 1 under BMM in three steps: 1) deriving the distribution of the discrepancy random variables, 2) writing the parameters of the batch's sum-of-discrepancies random variable from (2), and 3) assuming the M sum random variables form a Normal random vector (from the central-limit theorem), deriving a closed-form expression for the approximated success probability.

3.2.1 Deriving discrepancy probabilities. We first derive the distribution of the discrepancy under the BMM (without erroneous classifiers), as well as the joint probabilities between pairs of discrepancies. For simplicity of analysis we assume in this sub-section that M is even. We denote by $Z_m(\alpha, s)$ the probability that the outcome of a BM with parameter α and m classifiers has s correct classifiers, which gives $Z_m(\alpha, s) = \binom{m}{s}\alpha^s(1-\alpha)^{m-s}$, known as the s-th binomial term of order m and parameter α.

Let $Y_i \in \{0, 1\}$ be the random variable for classifier i's discrepancy under the BM model. From symmetry, the distribution of the discrepancy random variable is independent of i, so we use $p_Y \triangleq P(Y = 1)$ to denote it. Then, with the notation $M_r \triangleq M - r$, we have the following (proof omitted).

Proposition 3.1.

$$p_Y = \sum_{s=0}^{M/2-1} \left[\alpha Z_{M_1}(\alpha, s) + (1-\alpha) Z_{M_1}(1-\alpha, s)\right].$$

Denote by $p_{YY} \triangleq P(Y_i = 1, Y_j = 1)$ the probability that classifiers i and j ($i \neq j$) both have discrepancy equal to 1. Then we have the following (proof omitted).

Proposition 3.2.

$$p_{YY} = \sum_{s=0}^{M/2-2} \left[\alpha^2 Z_{M_2}(\alpha, s) + (1-\alpha)^2 Z_{M_2}(1-\alpha, s)\right] + 2\alpha(1-\alpha) Z_{M_2}(\alpha, M_2/2).$$

We need to derive similar discrepancy probabilities in the case of a single erroneous classifier, whose index we assume to be i'. We denote by $\widetilde{Y}_i$ the discrepancy random variable of index $i \neq i'$ (a non-erroneous classifier), and by $\widetilde{Y}'$ the discrepancy of the erroneous classifier i'. We denote by $\widetilde{Z}_m(\alpha, s)$ the probability that the outcome of a BM with parameter α and m classifiers, one of which is erroneous with parameter ϵ, has s correct classifiers. We have the following (proof omitted).

Proposition 3.3.

$$\widetilde{Z}_m(\alpha, s) = \binom{m-1}{s-1}\alpha'\alpha^{s-1}(1-\alpha)^{m-s} + \binom{m-1}{s}(1-\alpha')\alpha^s(1-\alpha)^{m-1-s},$$

where $\alpha' \triangleq \alpha(1-\epsilon) + (1-\alpha)\epsilon$ is the probability that the erroneous classifier is correct.

Denote $p_{\widetilde{Y}} \triangleq P(\widetilde{Y}_i = 1)$, which from symmetry is the same for any $i \neq i'$. Denote $p_{\widetilde{Y}'} \triangleq P(\widetilde{Y}_{i'} = 1)$. Then we have (proof omitted):

Proposition 3.4.

$$p_{\widetilde{Y}} = \sum_{s=0}^{M/2-1} \left[\alpha \widetilde{Z}_{M_1}(\alpha, s) + (1-\alpha)\widetilde{Z}_{M_1}(1-\alpha, s)\right],$$

$$p_{\widetilde{Y}'} = (1-\epsilon)p_Y + \epsilon(1-p_Y) = (1-2\epsilon)p_Y + \epsilon,$$

where p_Y is given in Proposition 3.1.

Moving to joint probabilities, denote $p_{\widetilde{Y}\widetilde{Y}} \triangleq P(\widetilde{Y}_i = 1, \widetilde{Y}_j = 1)$, which from symmetry is the same for any $i \neq j$ both of which $\neq i'$. Denote $p_{\widetilde{Y}\widetilde{Y}'} \triangleq P(\widetilde{Y}_i = 1, \widetilde{Y}_{i'} = 1)$, for some $i \neq i'$. Now we have (proof omitted):

Proposition 3.5.

$$p_{\widetilde{Y}\widetilde{Y}} = \sum_{s=0}^{M/2-2}\left[\alpha^2 \widetilde{Z}_{M_2}(\alpha, s) + (1-\alpha)^2 \widetilde{Z}_{M_2}(1-\alpha, s)\right] + 2\alpha(1-\alpha)\widetilde{Z}_{M_2}(\alpha, M_2/2),$$

$$p_{\widetilde{Y}\widetilde{Y}'} = \sum_{s=0}^{M/2-2} \left[\alpha\alpha' Z_{M_2}(\alpha, s) + (1-\alpha)(1-\alpha') Z_{M_2}(1-\alpha, s)\right] + \left[\alpha(1-\alpha') + (1-\alpha)\alpha'\right] Z_{M_2}(\alpha, M_2/2).$$

3.2.2 Distribution parameters of the batch's average discrepancies. Given a batch of N data points, each following the BM model independently, we wish to evaluate the probability that Detector 1 succeeds in declaring the erroneous classifier. This probability depends on the system parameters: M, α, ϵ, and also N. Detection success occurs when the erroneous classifier has a larger sum in (2) than *every* non-erroneous classifier. For analysis convenience, we replace the sum of (2) by the average over the batch, which is equivalent (the constant factor of N does not affect the detection). Define the random variable $\widetilde{F}_i = \frac{1}{N}\sum_{n=1}^{N} \widetilde{Y}_i(n)$ for a non-erroneous classifier i, and $\widetilde{F}' = \frac{1}{N}\sum_{n=1}^{N}\widetilde{Y}_{i'}(n)$ for the erroneous one i'. For any i, $\widetilde{F}_i$ has the same distribution, thus we sometime simplify its notation to $\widetilde{F}$. We then use the results of Section 3.2.1 to get the first and second moments (mean: μ, variance: σ^2, covariance: $\mathrm{cov}(\cdot,\cdot)$) of $\widetilde{F}$ and $\widetilde{F}'$:

$$\mu_{\widetilde{F}} = p_{\widetilde{Y}},\ \sigma^2_{\widetilde{F}} = \frac{p_{\widetilde{Y}} - p^2_{\widetilde{Y}}}{N},\ \mathrm{cov}(\widetilde{F}_i, \widetilde{F}_j) = \frac{p_{\widetilde{Y}\widetilde{Y}} - p^2_{\widetilde{Y}}}{N},$$

$$\mu_{\widetilde{F}'} = p_{\widetilde{Y}'},\ \sigma^2_{\widetilde{F}'} = \frac{p_{\widetilde{Y}'} - p^2_{\widetilde{Y}'}}{N},\ \mathrm{cov}(\widetilde{F}, \widetilde{F}') = \frac{p_{\widetilde{Y}\widetilde{Y}'} - p_{\widetilde{Y}}p_{\widetilde{Y}'}}{N},$$

where all the probabilities in the right-hand sides above are derived from BMM in Section 3.2.1.

3.2.3 Approximation with Joint Normal distribution. Inspired by the multivariate (Lindeberg-Feller) central limit theorem [15], we assume the random variables $\widetilde{F}_i$, $i \neq i'$ and $\widetilde{F}'$ to be jointly normal, even for N that is finite (but not too small). Denote by $\widetilde{\mathbf{F}}$ the $M \times 1$ random vector whose i-th element is $\widetilde{F}_i$, where in position i' it has the special random variable $\widetilde{F}'$. This random vector has a mean vector $\boldsymbol{\mu}_{\widetilde{F}}$ that has $\mu_{\widetilde{F}'}$ in position i' and $\mu_{\widetilde{F}}$ in all other positions. The covariance matrix Σ_N of this vector has the variances $\sigma^2_{\widetilde{F}}$ and $\sigma^2_{\widetilde{F}'}$ on the diagonal, and $\mathrm{cov}(\widetilde{F}_i, \widetilde{F}_j)$ and $\mathrm{cov}(\widetilde{F}, \widetilde{F}')$ off-diagonal. With the aforementioned assumption, we have that $\widetilde{\mathbf{F}} \sim \mathcal{N}\left(\boldsymbol{\mu}_{\widetilde{F}}, \Sigma_N\right)$, where $\mathcal{N}$ stands for the (multivariate) Normal distribution.

For convenient evaluation of the success probability, we apply the linear transform $\widetilde{\mathbf{D}} = \mathbf{C}\widetilde{\mathbf{F}}$, where $\mathbf{C}$ is a $(M-1)\times M$ matrix where $C_{i,i} = 1$ for all $i \in \{1, \ldots, i'-1\}$, $C_{i,i+1} = 1$ for all $i \in \{i', \ldots, M-1\}$, $C_{i,i'} = -1$ for all $i \in \{1, \ldots, M-1\}$, and zeros elsewhere. In words, $\mathbf{C}$ simply subtracts variable i' from every other variable, and removes the i'-th variable. As a result of this linear transformation, we get $\widetilde{\mathbf{D}} \sim \mathcal{N}\left(\mathbf{C}\boldsymbol{\mu}_{\widetilde{F}}, \mathbf{C}\Sigma_N\mathbf{C}^T\right)$. Note that $\widetilde{\mathbf{D}}$ has $M-1$ random variables that are identically distributed and equicorrelated, that is, they all have the same mean $\mu_{\widetilde{D}}$, the same variance $\sigma^2_{\widetilde{D}}$, and the same correlation $\rho_{\widetilde{D}}$ for every pair.

We now write the detection-success probability using $\widetilde{\mathbf{D}}$.

$$P\left(\begin{matrix}\text{Detector 1}\\ \text{success}\end{matrix}\right) = P\left(\hat{i} = i'\right) = P\left(\bigcup_{i \neq i'} \left\{\widetilde{F}_i < \widetilde{F}_{i'}\right\}\right)$$

$$= P\left(\bigcup_{i \neq i'} \left\{\widetilde{F}_i - \widetilde{F}_{i'} < 0\right\}\right) \approx P\left(\widetilde{\mathbf{D}} < \mathbf{0}\right),$$

where the < in the last argument is element-wise. The ≈ in the last equation captures the normal assumption. The last probability is called the multivariate-normal (negative) *orthant probability*.

Finally, we now approximate the orthant probability of $\widetilde{\mathbf{D}}$ using its quoted symmetries, ultimately yielding a closed-form expression. We define a Normal (univariate) random variable U with mean $\mu_u \triangleq \mu_{\widetilde{D}}/(\sigma_{\widetilde{D}}\sqrt{\rho_{\widetilde{D}}})$ and variance $\sigma_u^2 \triangleq \left(1 - \rho_{\widetilde{D}}\right)/\rho_{\widetilde{D}}$. We use $\phi(\cdot)$ and $\Phi(\cdot)$ to denote, respectively, the probability density function (PDF) and cumulative distribution function (CDF) of the standard zero-mean unit-variance Normal distribution. We also define the normal tail function $Q(x) \triangleq 1 - \Phi(x)$. Toward getting a closed-form expression, we use a 2-term approximation of $Q(\cdot)$ using exponential functions from [14]: $Q(x) \approx \hat{Q}(x) \triangleq \sum_{i=1}^{2} a_i e^{-b_i x^2}$, where $\{a_1, b_1, a_2, b_2\}$ are constants optimized in [14]. We can now write the final result (proof omitted).

Theorem 3.6.

$$P\left(\begin{matrix}\textit{Detector 1}\\ \textit{success}\end{matrix}\right) \approx \frac{\sigma_u}{\sqrt{2}} \sum_{s=0}^{M-1} \binom{M-1}{s} T(s), \tag{8}$$

where

$$T(s) = g\left(M-1, s\right) \Phi\left(-\frac{\beta}{\sqrt{2}\eta(M-1,s)}\right) +$$

$$(-1)^s \sum_{r=0}^{s} \binom{s}{r} g(s,r) \Phi\left(\frac{\beta}{\sqrt{2}\eta(s,r)}\right),$$

using the auxiliary parameters and functions $\beta \triangleq -\mu_u\sigma_u$, $\gamma \triangleq -\mu_u^2/2$, $\eta(m,n) \triangleq \sqrt{\sigma_u^2/2 + b_1(m-n) + b_2 n}$ *and* $g(u,v) = \frac{a_1^{u-v} a_2^{v}}{\eta(u,v)} e^{\frac{\beta^2}{4\eta^2(u,v)} + \gamma}$.

The accuracy of Theorem 3.6 in approximating the detection-success probability can be evidenced in Fig. 1 of Section 4.

3.3 Analysis of the Likelihood-Ratio Detector

We define a false-negative (FN) detection event to occur when a classifier i is erroneous (with parameter ϵ), but Detector 3 fails to detect that. This happens when $LLR_{\text{batch}-i} \leq \eta(M)$ while i is erroneous. Denote by $\widetilde{Q}$ any Q_i or $Q_{i,j}$ appearing in $LLR_{\text{batch}-i}$ in (6), and set $\tilde{p} \triangleq \widetilde{Q}/N$. We now write each component of $\widetilde{LLR}_{\text{batch}-i}$, where the notation $\tilde{}$ designates the erroneous case:

$$\tilde{l}(N,p) = N\tilde{p}\log\left(\frac{\tilde{p}}{p}\right) + N(1-\tilde{p})\log\left(\frac{1-\tilde{p}}{1-p}\right).$$

p is a known probability representing p_{X_i} or $p_{W_{i,j}}$. Next we expand Taylor's series for $f(\tilde{p}) = \tilde{p}\log\left(\frac{\tilde{p}}{p}\right)$ around a point $p' \triangleq a \cdot p + b$ and write

$$f(\tilde{p}) = f(ap+b) + f^{(1)}(ap+b)(\tilde{p} - ap - b) + O\left[(\tilde{p} - ap - b)^2\right],$$

where $f^{(1)}$ represent the first derivative. Neglecting terms of order 2 or more and setting $p' \triangleq (1-2\epsilon)p + \epsilon$ (equiv. $a = 1 - 2\epsilon$, $b = \epsilon$):

$$\tilde{l}(N,p) = Np'\log\left(\frac{p'}{p}\right) + N\left(\log\left(\frac{p'}{p}\right) + 1\right)(\tilde{p} - p')$$

$$+ N(1-p')\log\left(\frac{1-p'}{1-p}\right) - N\left(\log\left(\frac{1-p'}{1-p}\right) + 1\right)(\tilde{p} - p').$$

Neglecting the high terms is justified since the expected value of $\tilde{p}$ is p', so as $N \longrightarrow \infty$ the values $\tilde{p}$ concentrate around p'. With further simplification, we get

$$\tilde{l}(N,p) \approx Nl_1(p) + Nl_2(p)\tilde{p},$$

where $l_1(p) \triangleq \log\left(\frac{1-p'}{1-p}\right)$ and $l_2(p) \triangleq \log\left(\frac{1-p}{1-p'} \cdot \frac{p'}{p}\right)$. Summing over all terms in (6), we can write

$$\widetilde{LLR}_{\text{batch}-i} \approx Nl_1(p_{X_i}) + l_2(p_{X_i})\widetilde{Q_i} + \sum_{\substack{j=1\\ j\neq i}}^{M} \left[Nl_1(p_{W_{i,j}}) + l_2(p_{W_{i,j}})\widetilde{Q_{i,j}}\right] \triangleq \widetilde{V} + \sum_{j=1}^{M-1} \widetilde{W}_j. \tag{9}$$

Each of $\widetilde{V}$, $\{\widetilde{W}_j\}$ is a random variable that is a linear transformation of a corresponding binomial random variable $\widetilde{Q}$. The random variables $\widetilde{V}, \widetilde{W}_j, \widetilde{W}_{j'}, \ldots$ are dependent, due to the dependence between $\widetilde{X}_i, \widetilde{W}_{i,j}, \widetilde{W}_{i,j'}, \ldots$. In the limit $N \longrightarrow \infty$, we approximate $\widetilde{V}$,$\{\widetilde{W}_j\}$ to be Normally distributed (correlated) random variables.

We begin by writing the means of $\widetilde{V}$, $\{\widetilde{W}_j\}$

$$\mathbb{E}[\widetilde{V}] = Nl_1(p_{X_i}) + l_2(p_{X_i})\mathbb{E}[\widetilde{Q_i}] = Nl_1(p_{X_i}) + Nl_2(p_{X_i})p'_{X_i}$$

$$\mathbb{E}[\widetilde{W}_j] = Nl_1(p_{W_{i,j}}) + l_2(p_{W_{i,j}})\mathbb{E}[\widetilde{Q_{i,j}}] = Nl_1(p_{W_{i,j}}) + Nl_2(p_{W_{i,j}})p'_{W_{i,j}},$$

where $p'_{X_i} \triangleq (1-2\epsilon)p_{X_i} + \epsilon$ and $p'_{W_{i,j}} \triangleq (1-2\epsilon)p_{W_{i,j}} + \epsilon$. We can similarly derive the variances of $\widetilde{V}$, $\{\widetilde{W}_j\}$ (proofs omitted):

$$\mathrm{Var}(\widetilde{V}) = N\left(l_2(p_{X_i})\right)^2 p'_{X_i}(1 - p'_{X_i})$$

$$\mathrm{Var}(\widetilde{W}_j) = N\left(l_2(p_{W_{i,j}})\right)^2 p'_{W_{i,j}}(1 - p'_{W_{i,j}}),$$

as well as the covariances (omitted).

3.3.1 *Approximation with Normal random variable.*

We take the random variables defined in the right-hand side of (9) as a length-M vector $[\widetilde{V}, \widetilde{W}_1, \ldots, \widetilde{W}_{M-1}]$, and approximate it to be an M-variate Normal random vector with the parameters derived in the previous sub-section. We get that the distribution of $\widetilde{LLR}_{\text{batch}-i}$ – summing the vector elements – is approximately a Normal (uni-variate) random variable with mean $\mathbb{E}[\widetilde{V}] + \sum_{j=1}^{M-1}\mathbb{E}[\widetilde{W}_j]$ and variance $\mathrm{Var}(\widetilde{V}) + \sum_{j=1}^{M-1}\mathrm{Var}(\widetilde{W}_j) + 2\sum_{j=1}^{M-1}\mathrm{cov}(\widetilde{V}, \widetilde{W}_j) + 2\sum_{j<j'}\mathrm{cov}(\widetilde{W}_j, \widetilde{W}_{j'})$. Specializing this to the BMM, we get the following (proof omitted):

Proposition 3.7. *For any positive integer M, when the classifier outputs $\{X_i\}_{i=1}^{M}$ follow a BMM with parameters $\pi = 0.5$ and α, then the random variable $LLR_{\text{batch}-i}$ under H_1 (i erroneous with probability ϵ) can be asymptotically approximated as $\mathcal{N}(\mu_\epsilon, \sigma_\epsilon^2)$ with*

$$\mu_\epsilon = (M-1)N\left[l_1(p_W) + p'_W l_2(p_W)\right]$$

$$\sigma_\epsilon^2 = (M-1)N\left(l_2(p_W)\right)^2\left(p'_W(1-p'_W) + (M-2)cov_{W'}\right),$$

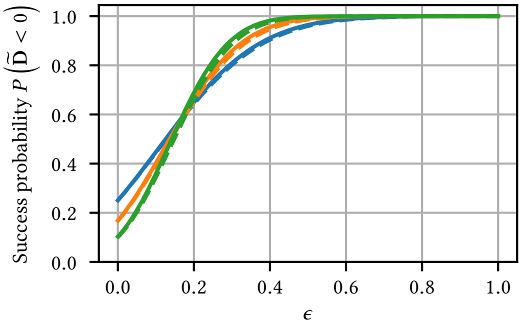

Figure 1: Analytical blind-detection success probability for $\alpha = 0.7, N = 200$, **and** $M = 4$ **(blue),** 6 **(orange),** 10 **(green).**

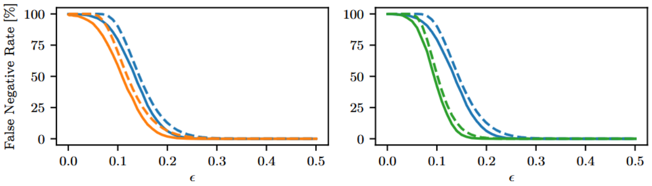

Figure 2: FN of BMM ensembles, simulated and analytical, for two detection thresholds (left) and two batch sizes (right).

with $p_W = \alpha^2 + (1-\alpha)^2, p'_W = (1-2\epsilon)p_W + \epsilon$ *and* $\mathrm{cov}_{W'} = (1-\epsilon)(\alpha^3+(1-\alpha)^3)+\epsilon((1-\alpha)\alpha^2+\alpha(1-\alpha)^2)-((1-2\epsilon)p_W+\epsilon)^2$.

Proposition 3.7 allows to calculate the approximate FN probability by simply evaluating the standard Normal CDF at the specified decision threshold: $\Phi((\eta(M)-\mu_\epsilon)/\sigma_\epsilon)$. The accuracy of this approximation is demonstrated in Fig. 2 of Section 4.

4 Numerical Results

Fig. 1 plots Detector 1's success probability under the joint-normal assumption, as a function of ϵ for $\alpha = 0.7$, $N = 200$, and three values of M. On the same plot we show both the exact numerical calculation (solid) and the closed-form approximation (8) (dashed). It can be seen that, as expected, the detection probability increases with ϵ, while for $\epsilon = 0$ it is roughly $1/M$. It can also be seen that for the higher values of ϵ, the performance improves as the ensemble size M is increased. The closed-form approximation is extremely close to the exact numerical values. Fig. 2 plots for $M = 5, \alpha = 0.8$, the FN probability of Detector 3 as a function of ϵ, comparing the empirical simulated FN rate (solid) to the analytical approximation (dashed) using Proposition 3.7. The left plot compares two values of detection threshold η. As expected, the lower η (blue) has smaller FN probability than the higher (orange). However, η also affects the false-positive probability (not shown here), in the reversed ordering. The right plot compares two values of batch size $N = 500$ (blue) and $N = 1000$ (green). Departing from the BM model toward practical setups, we implemented the classifiers using neural networks (NNs) trained for binary image classification. We independently trained $M = 4$ classifiers to distinguish between even and odd digits on the MNIST dataset [11]. At inference time, for each pair of N, ϵ values we sampled 10,000 N-batches from the test set, evaluated the NN-models on these inputs and applied the flipping by classifier i' chosen uniformly from $[M]$ in each N-batch. We recorded the classification accuracy after blocking the declared erroneous classifier on each N-batch. The results appear in Fig. 3. The plot depicts an average of 50 ensembles that were trained and evaluated independently. The figure compares the accuracy of Detector 1 (green) to a no-detection scheme (blue) that picks the label randomly in case of tie among the $M = 4$ classifiers. It also compares to an 'oracle' scheme (orange) that always blocks the true i'. It can be seen that Detector 1 approaches the oracle when ϵ is large enough, while in the no-detection scheme the accuracy drops linearly with ϵ.

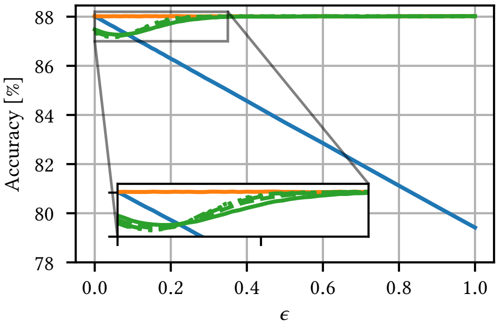

Figure 3: Experimental NN accuracy. $N = 100, 200, 300$ **in solid, dashed, and dotted green curves, respectively. Zoomed inset provided for the interesting region.**

References

[1] V. Barnett, *The study of outliers: Purpose and model*, Appl. Stat., vol. 27, no. 3, pp. 242–250, 1978.

[2] Y. Ben-Hur, A. Goren, D. Klang, Y. Kim and Y. Cassuto, *Ensemble classification with noisy real-valued base functions*, IEEE Journal on Selected Areas in Communications, vol. 41, no. 4, pp. 1067–1080, 2023.

[3] Y. Cassuto and Y. Kim, *Boosting for straggling and flipping classifiers*, 2021 IEEE International Symposium on Information Theory (ISIT), 2021.

[4] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. Wiley, 1973.

[5] Y. Freund and R. E. Schapire, *A decision-theoretic generalization of on-line learning and an application to boosting*, Journal of computer and System Sciences, vol. 55, no. 1, pp. 119–139, 1997.

[6] D. Hawkins, *Identification of Outliers*. Chapman & Hall, UK London, 1980.

[7] H. Hellstrom, J. M. Barros da Silva Jr., M. M. Amiri, M. Chen, V. Fodor, H. V. Poor and C. Fischione, *Wireless for Machine Learning: A Survey*. Foundations and Trends in Signal Processing, vol. 15, no. 4, pp. 290–399, 2022.

[8] A. Jain and A. Orlitsky,, *Optimal robust learning of discrete distributions from batches*, 2020 International Conference on Machine Learning (ICML), 2020.

[9] A. Juan and E. Vidal,, *Bernoulli mixture models for binary images*, 2004 IEEE International Conference on Pattern Recognition (ICPR), 2004.

[10] Y. Kim, J. Shin, Y. Cassuto and L. R. Varshney, *Distributed boosting classification over noisy communication channels*, IEEE Journal on Selected Areas in Communications, vol. 41, no. 1, pp. 141–154, 2023.

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. *Gradient-based learning applied to document recognition*. Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[12] E. Lehmann and J. Romano, *Testing Statistical Hypotheses, 4th edition*. Springer Cham, 2022.

[13] Y. Li, S. Nitinawarat and V. Veeravalli, *Universal outlier hypothesis testing*, IEEE Transactions on Information Theory, vol. 60, no. 7, pp. 4066–4082, 2014.

[14] I. M. Tanash and T. Riihonen, *Global minimax approximations and bounds for the Gaussian Q-function by sums of exponentials*, IEEE Transactions on Communications, vol. 68, no. 10, pp. 6514–6524, 2020.

[15] A. W. van der Vaart, *Asymptotic statistics*. Cambridge University Press, Cambridge, 2007.