Distributed Boosting Classification over Noisy Communication Channels

Yongjune Kim, Member, IEEE, Junyoung Shin, Yuval Cassuto, Senior Member, IEEE, and Lav R. Varshney, Senior Member, IEEE

Abstract—We address the design of inference-oriented communication systems where multiple transmitters send partial inference values through noisy communication channels, and the receiver aggregates these channel outputs to obtain a reliable final inference. Since large data items are replaced by compact inference values, these systems lead to significant savings of communication resources. In particular, we present a principled framework to optimize communication-resource allocation for distributed boosting classifiers. Boosting classification algorithms make a final decision via a weighted vote from the outputs of multiple base classifiers. Since these base classifiers transmit their partial inference values over noisy channels, communication errors would degrade the final classification accuracy. We formulate communication resource allocation problems to maximize the final classification accuracy by taking into account the importance of base classifiers and the resource budget. To solve these problems rigorously, we formulate convex optimization problems to optimize: 1) transmit-power allocations and 2) transmit-rate allocations. This framework departs from classical communication-systems optimizations in seeking to maximize the classification accuracy rather than the reliability of the individual communicated bits. Results from numerical experiments demonstrate the benefits of our approach.

Index Terms—Distributed inference, boosting, task-oriented communications, semantic communications, communication-resource optimization.

I. INTRODUCTION

In classical communications, information available to a transmitter is sent to a receiver through a channel, and the system’s objective is to reproduce this same information reliably at the receiver. With the advent of distributed machine learning and artificial intelligence (AI) applications, a communication system may have the objective of reliably reproducing an inference task on the information, rather than reproducing the information itself [2]–[4]. In this paper, we address the design of such a communication system where multiple transmitters send partial inference values through noisy channels, and the receiver aggregates these channel outputs to obtain a reliable final inference. Such a system offers significant advantages over classical communications: replacing large data items (e.g., images, videos) with compact inference values and also keeping the data items private from the receiver. This communication setting falls under the emerging concept of semantic communications (as well as the related task-oriented communications), where in this particular case the semantics come from the inference task at hand.

The inference task considered in this paper is binary classification, wherein each partial inference value sent by a transmitter is a binary value (i.e., a bit) output by a (weak) classification function run locally on the data item. Note, however, that reproducing these bits at the receiver is not the objective of the system. Rather, the system’s objective is to reproduce the final classification output aggregated from all the transmitters together. To study this problem in generality, we do not restrict the scope to a specific classification problem, and instead consider classifier functions obtained by state-of-the-art methods in ensemble machine learning. In ensemble approaches for classification, a set of base classifiers are trained (jointly or separately), and the final classification is obtained from the base classifiers’ binary outputs by weighted voting. Our objective in this paper is to design the communication schemes to carry partial classifications so as to maximize the reliability of the final classification. The approach we take is resource allocation, and the reliability measure we adopt is minimizing the probability of a mismatch between the noisy and noiseless final classifications, which we call the mismatch probability.

The most powerful known ensemble method is boosting, by which the base classifiers are trained jointly, and the weights assigned to classifiers reflect their classification quality on the training set [5]. Boosting can achieve good classification accuracy even if the base classifiers have a performance that is only slightly better than random guessing [6], [7]. Adaptive boosting (AdaBoost) is the most widely used form of boosting [6], [8]; it works well for classification problems such as face detection [9] and can be extended to regression problems [10].

For a binary classification (i.e., inference) problem, the final output (prediction) of an AdaBoost classifier is given by

\[ f(x) = \text{sign} \left( \sum_{t=1}^{T} \alpha_t f_t(x) \right), \]  

(1)
where $\mathbf{x}$ denotes the input data and $\alpha_t$ denotes the weight coefficient for the base classifier $f_t(\cdot)$. AdaBoost assigns larger weight coefficients to more accurate (or important) base classifiers during training [7], [8]. Unlike AdaBoost, in an alternative scheme called bagging, the weight coefficients of the base classifiers are uniform (i.e., $\alpha_t = \frac{1}{t}$ for all $t \in \{1, \ldots , T\}$) and the output corresponds to majority voting [11]. It is commonly held that boosting has better performance than bagging, and the non-uniformity of the $\alpha_t$ coefficients is certainly a key feature contributing to its strength. In addition to the classical AdaBoost, our approach works more generally with any boosting algorithms, including LPBoost [12], BrownBoost [13], MadaBoost [14], and gradient boosting [10].

In our system setup, the base classifiers of AdaBoost are spatially distributed from the aggregator (weighted voter) and must transmit their binary outputs $f_t(\mathbf{x})$ over noisy channels as shown in Fig. 1. Here, $\mathbf{x}$ is a data item to be classified; for simplicity, we mark in Fig. 1 the full $\mathbf{x}$ as the input to all base classifiers, but in practice, each classifier may operate only on a subset of the features of $\mathbf{x}$. The noise $\mathbf{z} = (z_1, \ldots , z_T)$ results in communication errors between the base classifiers and the aggregator. Alternatively, $\mathbf{z}$ can originate from noise in the computation hardware of base classifiers [15], [16]. We assume the aggregator is implemented in a noiseless manner. Within this model, we develop a principled framework to optimize the classification accuracy by allocating system resources to base classifiers according to their importance prescribed in the weight-coefficient vector $\alpha$.

The impact of $\mathbf{z}$ can be controlled by allocating system resources such as transmit power and transmit rate. Ideally, the system resources should be allocated to minimize the classification-error probability. Since the classification-error probability of boosting depends on the data sets and on the base classifiers and their training algorithms (which we do not control), we instead use the mismatch probability as a performance metric that is better related to the system resources. The mismatch probability has clear operational importance as an upper bound on the increase in classification-error probability due to communication noise. Toward tractable minimization of the mismatch probability, we define three proxies and minimize them instead of the mismatch probability itself. These proxies are: 1) Markov proxy, 2) Chernoff proxy, and 3) Gaussian proxy. We formulate optimization problems to minimize these proxies for a given resource budget. Using the Markov and Chernoff proxies, we can minimize upper bounds on the classification-error probability of noisy inference. The upper bounds are derived by Markov’s inequality and the Chernoff bound, respectively. Additionally, the Gaussian proxy enables the minimization of an estimate of the classification-error probability based on a Gaussian approximation of the noise added to the sum in (1) due to erroneous inputs from base classifiers. This approach of minimizing upper bounds and approximations is effective in many engineering problems, e.g., [17]–[19]. The first two proxies: Markov and Chernoff, are provable upper bounds on the mismatch probability, so their minimization has the theoretical value of providing tight formal upper bounds on the mismatch probability (which in turn provide upper bounds on the classification-error probability). The third, the Gaussian proxy, does not give formal upper bounds on the mismatch probability, but its minimization is shown empirically to give the lowest mismatch probabilities.

In particular, we formulate convex optimization problems to optimize: 1) transmit-power allocations for uncoded communications and 2) transmit-rate allocation for coded communications. In both cases, we adopt widely used canonical models: binary signals in additive Gaussian noise in the former and compressed binary streams over rate-limited bit pipes under Hamming distortion in the latter. We show that the proposed framework can effectively reduce the classification-error probability for a given transmit power/rate budget. Further, we derive analytic solutions to these optimization problems based on the Karush-Kuhn-Tucker (KKT) conditions. The resource allocation problems owe their efficient and analytical solutions via convex optimization to the unique property of AdaBoost having non-negative coefficients $\alpha_t$ for $t \in \{1, \ldots , T\}$.

The resulting inference-oriented optimization problems are then formulated as water-filling schemes, similar in spirit to Shannon’s water-filling schemes in classical communications [20]. Unlike classical water-filling where all transmitted bits are treated equally, we take into account the importance of the transmitted bits to the final classification output, and our objective is to improve the inference (classification) accuracy instead of to recover the individual transmitted bits for given resource constraints. This formulation allows characterizing trade-offs between communication resources and inference accuracy, as we demonstrate in empirical evaluations.

Beyond noisy communication, our framework also applies to noisy computation. For example, the quality of computations on noisy hardware can be changed by controlling supply voltage [15], replicating computations [21], [22], and implementing granular quantization [18]. The proposed framework enables optimizing these system resources of boosting classifiers in a principled manner.

Most of the related work in the machine-learning field focuses on mitigating noise during training, and not in the inference stage. Noise in the form of mislabelled training sets, in a model called random classification noise (RCN), assumes each label in the training set is flipped with some probability [23], [24]. Several studies have investigated the behavior of AdaBoost under label noise and proposed more robust training algorithms [14], [25], [26]. Label noise is complementary to the problem of noisy inference we consider.
in this paper. Other works consider noisy computations of the base-classifier functions while assuming that the same noisy hardware is available during training [16], [27]; this is contrary to our framework in which training is assumed to be performed independent of the noise.

In the areas of semantic and task-oriented communications, several works are relevant to the proposed framework. Distributed detection by multiple sensors [28]–[30] (mostly studied in the signal-processing field) assumes some known distribution of input data \( x \) and aims to optimize the thresholds used by sensors to obtain partial decisions under statistical aggregation rules such as the likelihood ratio test (LRT). In contrast, our problem does not require knowing the distribution of \( x \), and the base classifiers come from rich machine learning models, including decision trees [9], [25], support vector machines [31], and neural networks [32], [33]. More importantly, we focus on optimizing communication resources rather than assuming that the communication resources are given. Unlike the distributed detection problems, the trained importance coefficients of base classifiers play a critical role in our optimization problems. Note that the trained importance coefficients quantify the significance and usefulness of the transmitted information. In [2], the significance of information is regarded as the semantics of information.

More broadly, our work is related to the problem of task-oriented source coding, for example, distributed hypothesis testing over a noisy channel [34], in which the channel code and the transmitter’s decisions are designed jointly. In our work, joint source-channel coding is manifested in varying distortion values assigned to base classifiers, based on the optimized channel resources given to each classifier.

The contributions of this paper can be summarized as follows. 1) Defining a new model for distributed inference over noisy channels with a performance metric – the mismatch probability – that allows optimized communication for state-of-the-art boosting algorithms (Section II). 2) Prescribing three optimization proxies for tractable optimization of the mismatch probability (Section III). 3) Formulating rigorous convex optimization problems for minimizing the proxies under various resource constraints from realistic communication scenarios (e.g., Gaussian channels, fading channels), and 4) deriving efficient solutions for the convex problems via KKT conditions, water-filling, and iterative methods (Section IV). 5) An empirical study of the proposed methods, using real-world data sets, showing their advantages over non-optimized communication resources (Section V).

II. NOISY INFERENCE MODEL WITH ADABOOST

We present in this section the model of noisy ensemble classification. We choose to present it using the AdaBoost ensemble method for concreteness, but other weighted-ensemble methods can be employed without change to our results.

A. AdaBoost

Consider the standard supervised classification problem, for the given training set of labeled data points \( S = \{(x_1, y_1), \ldots, (x_N, y_N)\} \). The objective of training is to estimate the unknown function \( f(x) \) based on the given training set \( S \). The input vector is given by \( x_n = (x_{n,1}, \ldots, x_{n,D}) \), for \( n \in \{1, \ldots, N\} \), where \( D \) denotes the dimension of the input vectors. The output variables \( y_n \), for \( n \in \{1, \ldots, N\} \), are typically drawn from a discrete set of classes, i.e., \( y_n \in \{1, \ldots, K\} \), where \( K \) denotes the number of classes. For a binary classification problem, we assume \( y_n \in \{+1, -1\} \).

AdaBoost trains the base classifiers in sequence to minimize an exponential error function [7], [8]. Each base classifier is trained using a weighted form of the training set in which the data weights \( w = (w_1, \ldots, w_N) \) depend on the performance of previous base classifiers. In particular, data points that are misclassified by one of the base classifiers are given greater weight when used to train the next base classifier. Once all base classifiers have been trained, their outputs are combined through weighted voting [7].

The data weights \( w = (w_1, \ldots, w_N) \) are distinct from the classifier weight coefficients \( \alpha = (\alpha_1, \ldots, \alpha_T) \). AdaBoost determines both \( w \) and \( \alpha \) during training (see Algorithm 1). Once training is done, only the coefficients \( \alpha \) are used to classify new data points during the inference stage.

Algorithm 1 Training of AdaBoost for binary classification [7]

1: Initialize the data weights \( w^{(1)} \) by setting \( w_n^{(1)} = \frac{1}{N} \) for all \( n \in \{1, \ldots, N\} \).
2: for \( t = 1 : T \) do
3: \hspace{1em} Train a base classifier \( f_t(x) \) by minimizing the weighted error function
4: \hspace{2em} \( J_t = \sum_{n=1}^{N} w_n^{(t)} I(f_t(x_n) \neq y_n) \), \hspace{2em} (2)
5: \hspace{2em} where the superscript \( (t) \) denotes the iteration index for the data weights \( w \) and \( I(\cdot) \) denotes the indicator function, i.e., \( I(f_t(x_n) \neq y_n) \) equals 1 if \( f_t(x_n) \neq y_n \) and 0 otherwise.
6: \hspace{2em} Evaluate
7: \hspace{3em} \( \varepsilon_t = \frac{\sum_{n=1}^{N} w_n^{(t)} I(f_t(x_n) \neq y_n)}{\sum_{n=1}^{N} w_n^{(t)}} \). \hspace{2em} (3)
8: \hspace{2em} Compute the classifier coefficients
9: \hspace{3em} \( \alpha_t = \log \frac{1 - \varepsilon_t}{\varepsilon_t} \). \hspace{2em} (4)
10: \hspace{2em} Update the data weights
11: \hspace{3em} \( w_n^{(t+1)} = w_n^{(t)} \exp \{\alpha_t I(f_t(x_n) \neq y_n)\} \). \hspace{2em} (5)
12: end for
13: return the trained base classifiers \( \{f_t(\cdot)\} \) and the corresponding coefficients \( \alpha \) for \( t \in \{1, \ldots, T\} \).

The final AdaBoost model is given by (1) where the base classifiers and coefficients are decided by Algorithm 1. The classification-error probability of the trained model \( f(\cdot) \) is

\[
P_{e,f} = \Pr(f(x) \neq y),
\]

where \( y \) is the true label corresponding to \( x \).
\textbf{Remark 1 (Positive Coefficients):} If a base classifier is better than random guessing, then $\alpha_t > 0$ for any $t \in \{1, \ldots, T\}$ [6].

\textbf{Remark 2 (Normalized Coefficients):} We normalize the coefficients such that $\sum_{t=1}^{T} \alpha_t = 1$. Notice that normalization does not affect the classification output in (1).

\textbf{Remark 3 (Distinction from Distributed Detection Problem):} We note that $\epsilon_t$ in (3) depends on the data weights $w$ unlike the distributed detection problem in [28].

\section*{B. Noisy Inference with AdaBoost}

Suppose that the base classifiers’ outputs may be flipped due to the noise $z_t$, i.e., $f_t(x) \neq \hat{f}_t(x)$, where

$$\hat{f}_t(x) = \text{sign}(s_t + z_t),$$

where $\text{sign}(x) = -1$ for $x < 0$ and $\text{sign}(x) = 1$ for $x \geq 0$. We suppose that $f_t(x)$ is transmitted using a symbol $s_t$. The mismatch event of the $t$-th base classifier is denoted by

$$\delta_t = I(f_t(x) \neq \hat{f}_t(x)).$$

Then, we can define the base classifiers’ mismatch probabilities as $p = (p_1, \ldots, p_T)$ where

$$p_t \triangleq \Pr(f_t(x) \neq \hat{f}_t(x)) = \mathbb{E}[\delta_t].$$

In the sequel, the expectation over the distribution of $x$ will be replaced by the empirical mean over the given data set.

The final output of noisy AdaBoost is given by

$$\hat{f}(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t \hat{f}_t(x)\right).$$

Then, the final mismatch probability (i.e., mismatch probability of the final output) is given by

$$P_m \triangleq \Pr(\hat{f}(x) \neq \hat{f}(x)),$$

which captures the deleterious impact of $z$ on the final classification accuracy. We can expect that the final mismatch probability $P_m$ depends on the base classifiers’ mismatch probabilities $p = (p_1, \ldots, p_T)$.

The classification-error probability of the noisy AdaBoost is upper-bounded as

$$P_e = P_{e,f} \leq P_{e,f} + P_m,$$

where $P_{e,f}$ denotes the classification-error probability of noise-free AdaBoost. In [18], [19], the mismatch probability characterizes the impact of quantization noise due to limited bit precision. Note that $P_{e,f}$ solely depends on the AdaBoost algorithm and the dataset, i.e., $P_{e,f}$ is independent of $z$. Hence, we focus on the final mismatch probability $P_m$ to reduce the detrimental impact of the noise $z = (z_1, \ldots, z_T)$.

\section*{III. Importance Metrics of Base Classifiers}

We define three proxies to the mismatch probability: 1) Markov proxy, 2) Chernoff proxy, and 3) Gaussian proxy. These proxies induce different importance metrics of base classifiers. We provide theoretical justification for the proxies, i.e., the relation between the mismatch probability and the proxies.

\subsection*{A. Markov Proxy}

Let us define the \textit{Markov proxy}, which comes from Markov’s inequality.

\textbf{Definition 1 (Markov Proxy):} The Markov proxy $\hat{p}_M$ of noisy AdaBoost is given by

$$\hat{p}_M = \sum_{t=1}^{T} \alpha_t p_t,$$

which is the nonnegative weighted sum of $p_t$.

We derive an upper bound on the mismatch probability $P_m$ based on Markov’s inequality and show that this upper bound can be lowered by minimizing the Markov proxy $\hat{p}_M$.

\textbf{Theorem 1 (Upper Bound by Markov’s Inequality):} The mismatch probability of $x_n$ is upper bounded as follows:

$$P_m(x_n) \leq \frac{2\hat{p}_M}{\gamma_n},$$

where

$$\gamma_n = \left| \sum_{t=1}^{T} \alpha_t f_t(x_n) \right|,$$

which represents the decision margin of $x_n$. Then, an upper bound on the mismatch probability $P_m$ is given by

$$P_m \leq \left( \frac{1}{N} \sum_{n=1}^{N} \frac{2}{\gamma_n} \right) \cdot \hat{p}_M.$$

\textbf{Proof:} The proof is given in Appendix A. \hfill \blacksquare

\textbf{Remark 4:} Because of normalized condition in Remark 2, $\gamma$ is bounded between $0 \leq \gamma \leq 1$. Note that $\gamma$ is a metric related to confidence of classification rather than classification accuracy itself.

Higher decision margin $\gamma_n$ and/or lower Markov proxy $\hat{p}_M$ reduce the upper bound on the mismatch probability. The margin $\gamma_n$ depends on the input vector $x_n$ and the trained model. In contrast, $\hat{p}_M$ depends on $z$, whose distribution we can control by resource allocation; hence minimizing $\hat{p}_M$ is pursued in Section IV.

\textbf{Remark 5:} For a given dataset and trained model, the upper bound (14) depends only on the Markov proxy $\hat{p}_M$. Hence, our objective is to minimize $\hat{p}_M$ by controlling $p = (p_1, \ldots, p_T)$.

\subsection*{B. Chernoff Proxy}

In this subsection, we define the \textit{Chernoff proxy} via the Chernoff bound.

\textbf{Definition 2 (Chernoff Proxy):} The Chernoff proxy $\hat{p}_C$ of the mismatch probability is given by

$$\hat{p}_C(s) = \sum_{t=1}^{T} (e^{\alpha_t s} - 1) p_t,$$

where $s > 0$ is a parameter of the Chernoff bound.

\textbf{Remark 6:} Since $\alpha_t > 0$ and $s > 0$, $e^{\alpha_t s} - 1 > 0$.

We derive an upper bound on the mismatch probability $P_m$ from the Chernoff bound and show that this upper bound can be reduced by minimizing the Chernoff proxy $\hat{p}_C$. 
**Theorem 2 (Upper Bound by Chernoff Bound):** The mismatch probability is upper-bounded as follows:

\[ P_m \leq \mathbb{E} \left[ \exp \left( -s \cdot \frac{\gamma_n}{2} \right) \right] \cdot \exp \left( \tilde{\mu}_C(s) \right), \tag{18} \]

for any \( s > 0 \). Notice \( \mathbb{E} \left[ \exp \left( -s \cdot \frac{\gamma_n}{2} \right) \right] \) can be calculated by

\[ \mathbb{E} \left[ \exp \left( -s \cdot \frac{\gamma_n}{2} \right) \right] = \frac{1}{N} \sum_{n=1}^{N} \exp \left( -s \cdot \frac{\gamma_n}{2} \right). \tag{19} \]

**Proof:** The proof is given in Appendix B. This upper bound can be tightened by minimizing the Chernoff proxy \( \tilde{\mu}_C \). Like the Markov proxy, a higher decision margin \( \gamma_n \) decreases the upper bound.

It is worth mentioning that \( s \) should be carefully chosen because of a trade-off between \( \mathbb{E} \left[ \exp \left( -s \cdot \frac{\gamma_n}{2} \right) \right] \) and \( \tilde{\mu}_C \). A smaller \( s \) decreases \( \tilde{\mu}_C \) while increasing \( \mathbb{E} \left[ \exp \left( -s \cdot \frac{\gamma_n}{2} \right) \right] \).

Since the optimal \( s \) and \( p \) are interdependent, we propose an iterative algorithm to jointly optimize \( s \) and \( p \) (see Algorithm 2 in Section IV).

**C. Gaussian Approximation**

As in Definition 1 and Definition 2, we define a third proxy \( \tilde{\mu}_G \) of the mismatch probability is given by

\[ \tilde{\mu}_G = \sum_{t=1}^{T} \alpha_t^2 p_t. \tag{20} \]

Suppose that \( f(x_n) = \text{sign}(g(x_n)) \) where \( g(x_n) = \sum_{t=1}^{T} \alpha_t \tilde{f}_t(x_n) \). Also, we set \( \tilde{f}(x_n) = \text{sign}(\tilde{g}(x_n)) \) where \( \tilde{g}(x_n) \) is given by

\[ \tilde{g}(x_n) = \sum_{t=1}^{T} \alpha_t \tilde{f}_t(x_n) \]
\[ = \sum_{t \in T^+_n} \alpha_t \tilde{f}_t(x_n) + \sum_{t \in T^-_n} \alpha_t \tilde{f}_t(x_n) \]
\[ = \sum_{t \in T^+_n} \alpha_t(1 - \delta_{t,n}) + \sum_{t \in T^-_n} \alpha_t(-1 + 2\delta_{t,n}) \tag{21} \]
\[ = g(x_n) + v_n, \tag{22} \]

where \( T^+_n = \{ t \mid f_t(x_n) = 1 \} \) and \( T^-_n = \{ t \mid f_t(x_n) = -1 \} \), respectively. Note that \( \delta_{t,n} = I(f_t(x_n) \neq \tilde{f}_t(x_n)) \).

We set \( g(x_n) = \pm \gamma_n \) and \( v_n \) in (22) as the signal term and the noise term, respectively. The noise term \( v_n \) is given by

\[ v_n = -2 \left( \sum_{t \in T^+_n} \alpha_t \delta_{t,n} - \sum_{t \in T^-_n} \alpha_t \delta_{t,n} \right). \tag{23} \]

**Theorem 3:** The noise term \( v_n \) for \( n \in \{ 1, \ldots, N \} \) can be modeled as a Gaussian distribution, i.e., \( v_n \sim \mathcal{N}(\mu_v, \sigma_v^2) \) by the central limit theorem. Then,

\[ \mu_v = -2 \sum_{t \in T^+_n} \alpha_t p_t - \sum_{t \in T^-_n} \alpha_t p_t, \tag{24} \]
\[ \sigma_v^2 = 4 \sum_{t=1}^{T} \alpha_t^2 p_t (1 - p_t). \tag{25} \]

**Proof:** The proof is given in Appendix C. Observe that the variance in (25) is data independent, and thus its minimization by resource allocation is an effective way to reduce the mismatch probability.

**Remark 7:** If \( p_t \ll 1 \) (i.e., \( p_t^2 \ll p_t \)), then \( \sigma_v^2 \approx 4 \tilde{\mu}_G \). The advantage of Gaussian proxy is the convexity of the right-hand side of (20) (see Section IV).

Based on the Gaussian approximation, we can derive an estimate of the mismatch probability.

**Corollary 1:** An estimate of the mismatch probability is given by

\[ P_m(x_n) \approx Q \left( \frac{\gamma_n - |\mu_v|}{2 \sqrt{\sigma_v}} \right), \tag{26} \]

where \( Q(x) = \frac{1}{\sqrt{2\pi}} \int_{x}^{\infty} \exp \left( -\frac{u^2}{2} \right) du \).

The minimum \( |\mu_v| \) and the minimum \( \tilde{\mu}_G \) are desired to reduce the estimate of mismatch probability. While \( \mu_v \) depends on the input vector \( x_n \), the Gaussian proxy depends only on the trained \( \alpha \) and the base classifiers’ mismatch probability \( p = (p_1, \ldots, p_T) \). Hence, we can minimize the Gaussian proxy to reduce the mismatch probability.

Note that each of the three proxies can be described by

\[ \sum_{t=1}^{T} \beta_t p_t, \tag{27} \]

where \( \beta_t \) denotes the importance metric of the \( t \)-th base classifier. Table I lists the importance metrics for the three proxies.

![Table I](https://example.com/table1.png)

**IV. RESOURCE ALLOCATION FOR NOISY CLASSIFICATION**

**A. Formulation of Optimization Problems**

We investigate optimization approaches to determine the optimal \( p = (p_1, \ldots, p_T) \) for a given resource constraint. By optimizing the proposed proxies, we attempt to reduce the mismatch probability, i.e., reduce the noise impact on classification accuracy.

An important assumption is that the mismatch probabilities of base classifiers can be controlled by resource allocation.
Suppose that the mismatch probability of the $t$-th base classifier $p_t$ can be described by resource $r_t$, i.e., $p_t = p(r_t)$. Then, we can formulate the following optimization problem for a given resource budget $C$:

$$\begin{align*}
\text{minimize} & \quad \sum_{t=1}^{T} \beta_t p(r_t) \\
\text{subject to} & \quad \sum_{t=1}^{T} c(r_t) \leq C,
\end{align*}$$

(28)

where the objective function depends on the importance metric $\beta = (\beta_1, \ldots, \beta_T)$. Also, $c(r_t)$ denotes the cost of the allocated resource to the $t$-th base classifier.

Remark 9: If $p(r_t)$ and $c(r_t)$ are convex, then the optimization problem (28) is also convex because $\beta_t$ is positive for all $t \in \{1, \ldots, T\}$ in any of the three proxies (Remark 8). For the Markov proxy and the Gaussian proxy, we can obtain the optimal resource allocation by solving (28) directly using convex programming.

We propose an iterative algorithm to jointly optimize $s$ and $p$ (see Algorithm 2). Algorithm 2 attempts to minimize the upper bound (18) by alternating between optimizations of (29) and (30). Since the upper bound (18) holds for any $s > 0$, we need to optimize $s$ as well as $p$ to make the upper bound tight. We start with an arbitrary point $s(0) \in (0, \infty)$, optimize $p^{(1)}$, then find the optimal $s^{(1)}$ given $p^{(1)}$ and iterate. Notice that the choice of $s$ does not affect the feasible set, so the optimization of $s$ in (30) can be done in closed form without worry that $p$ would become infeasible. Step 1 and Step 2 find $p^{(i+1)}$ minimizing $\tilde{p}_C(s^{(i)})$ for a given $s^{(i)}$, which is a convex problem for any $s^{(i)} > 0$. It is valid because $p$ affects only $\tilde{p}_C(s)$ among the upper bound of (18), which corresponds to (28). Step 3 finds the optimal $s^{(i+1)}$ minimizing the upper bound of (18) for a given $p^{(i+1)}$. Step 4 introduces a small positive $\epsilon > 0$ to satisfy the condition of $s > 0$.

We show that the upper bound is a convex function of $s$ and the solution of (30) is optimal.

Theorem 4: For given $p = (p_1, \ldots, p_T)$ and $\gamma = (\gamma_1, \ldots, \gamma_N)$, the upper bound of (18) is a function of $s$ as follows:

$$h(s) = \frac{1}{N} \sum_{n=1}^{N} e^{-s \\frac{2}{\gamma}} \exp \left( \sum_{t=1}^{T} (e^{s\alpha_t} - 1) p_t \right),$$

(31)

which is convex; hence $s$ satisfying $h'(s) = 0$ (i.e., (30)) is optimal.

Proof: The proof is given in Appendix D.

Corollary 2: If $\gamma_n = \gamma_0$ for all $n \in \{1, \ldots, N\}$ and $\alpha_t = \frac{1}{T}$ for all $t \in \{1, \ldots, T\}$, then the optimal $s$ is given by

$$s^* = \max \left\{ T \cdot \log \frac{\gamma_0}{2 \tilde{p}}, \epsilon \right\},$$

(32)

where $\tilde{p} = \frac{\sum_{t=1}^{T} p_t}{T}$.

Proof: If $\gamma_n = \gamma_0$ for any $n$ and $\alpha_t = \frac{1}{T}$ for any $t$, then (31) is given by

$$h(s) = \exp \left( -\frac{\gamma_0}{2} s + \sum_{t=1}^{T} (e^{s} - 1) p_t \right).$$

(33)

Then, the minimization of $h(s)$ is equivalent to minimizing $q(s) = -\frac{\gamma_0}{2} \cdot s + \sum_{t=1}^{T} (e^{s} - 1) p_t$.

It is clear that $s^* = T \cdot \log \frac{2}{\gamma_0 \tilde{p}}$ satisfies $q'(s) = 0$. If $s^* \leq 0$, then we set $s^* = \epsilon$ to prevent $s^* \leq 0$.

We observe that a larger noise margin $\gamma_0$ increases the optimal $s$ whereas a larger $\tilde{p}$ reduces the optimal $s$. In the following subsections, we consider two optimization problems for boosting classifiers over noisy channels.

B. Transmit Power Optimization for Uncoded Communications

We formulate a transmit power optimization problem over $T$ parallel channels as in classical water-filling [20]. Here, the objective is minimizing the mismatch probability, whereas classical water-filling maximizes the channel capacity.

Suppose that $f_t(x) \in \{+1, -1\}$ is transmitted using a symbol from $\{s_t, -s_t\}$, which is corrupted by the noise $z_t \sim N(0, \sigma^2)$ as shown in Fig. 1. Then, the $t$-th base classifier's
mismatch probability can be defined by the communication error probability over the Gaussian channel as follows:

$$p_t = Q \left( \sqrt{SNR_t} \right) = Q \left( \frac{s_t}{\sigma_t} \right), \quad (34)$$

where the signal-to-noise ratio (SNR) is $SNR_t = \frac{s_t^2}{\sigma_t^2}$. Hence, $p_t$ can be controlled by allocating transmit power $s_t^2$. Then, the optimization problem (28) will be as follows:

$$\begin{align*}
\text{minimize} & \quad \sum_{t=1}^{T} \beta_t Q \left( \frac{s_t}{\sigma_t} \right) \\
\text{subject to} & \quad \sum_{t=1}^{T} s_t^2 \leq C \\
& \quad s_t \geq 0, \quad t = 1, \ldots, T,
\end{align*} \tag{35}$$

where $C$ represents the total power budget.

**Remark 10:** The power allocation problem (35) is convex since $p_t = Q \left( \frac{s_t}{\sigma_t} \right)$ is convex for $s_t \geq 0$, i.e., $\frac{d^2 Q(x)}{dx^2} = \frac{2}{\sqrt{\pi e x^2}} \exp \left( -\frac{x^2}{2} \right) \geq 0$ for $x \geq 0$.

**Theorem 5:** The optimal solution $s^*$ of (35) satisfies the following condition:

$$SNR^*_t = \left( \frac{s^*_t}{\sigma_t} \right)^2 = W \left( \frac{1}{8 \pi \nu^2} \cdot \frac{\beta_t^2}{\sigma_t^4} \right), \quad (36)$$

where $\nu$ is a positive dual variable of the KKT conditions. Also, $W(\cdot)$ denotes the Lambert $W$ function (i.e., the inverse function of $f(x) = xe^x$) [35].

**Proof:** The proof is given in Appendix E. \hfill \blacksquare

Since $W(x)$ is an increasing function for $x \geq 0$, (36) shows that we allocate higher SNR for higher $\beta_t$ as shown in Fig. 3. For a classifier with $\beta_t \approx 0$, the corresponding SNR is $SNR^*_t \approx 0$ because $W(0) = 0$.

**Corollary 3:** If $\sigma_t = \sigma$ for all $t \in \{1, \ldots, T\}$, then the optimized proxy of (35) can be approximated as:

$$\begin{align*}
\sum_{t=1}^{T} \beta_t Q \left( \frac{s_t}{\sigma_t} \right) & \approx \frac{T}{2} \exp \left( -\frac{C}{2T\sigma^2} \right) \left( \prod_{t=1}^{T} \beta_t \right)^{\frac{1}{T}} \cdot (37)
\end{align*}$$

where $\left( \prod_{t=1}^{T} \beta_t \right)^{\frac{1}{T}}$ is the geometric mean of $\beta$.

**Proof:** The proof is given in Appendix F. \hfill \blacksquare

A smaller geometric mean of $\beta$ implies a lower proxy value. Note that increasing SNR $\frac{C}{\nu}$ decreases the proxy value.

**Remark 11:** The geometric mean of $\beta$ is maximized by uniform $\alpha = (\frac{1}{T}, \ldots, \frac{1}{T})$. Thus the non-uniform coefficients of AdaBoost’s classifiers contribute to lower classification error probability. This suggests an advantage of AdaBoost over bagging that assigns the same coefficients to all classifiers, i.e., $\alpha = (\frac{1}{T}, \ldots, \frac{1}{T})$. This is noteworthy because AdaBoost is known to be less robust than bagging in the face of noisy data labels [24], [25].

### C. Transmit Rate Optimization for Coded Communications

In this subsection, we investigate the relationship between inference accuracy and transmission rate, which is an important open question in semantic communications [4]. We consider using channel coding to protect the transmitted classifier outputs $f_t(x)$. In this setting, we assume each classifier generates a batch of output bits and encodes them into a codeword which it then sends over a channel with a certain capacity. The optimization problem is now allocating the capacities of the individual channels, subject to a total capacity budget. By the channel coding theorem, we can attain an arbitrarily small probability of communication error if the rate input to the channel is below the capacity. Hence in our setup, we match each classifier rate to the capacity allocated to its channel by introducing distortion.

To this end, we model each classifier initially as a source of rate 1 bit per classification output and then introduce its outputs with Hamming distortion to fit its allocated communication rate. According to rate-distortion theory [36], a rate 1 source can be compressed to the following rate:

$$r_t = 1 - h(p_t), \quad (38)$$

where the Hamming distortion of $p_t$ (for $0 \leq p_t \leq \frac{1}{2}$) corresponds to the expected proportion of bit errors and $h(\cdot)$ is the binary entropy function. The appeal of this setup is that it allows optimizing the rate allocations without considering the specifics or even types of channels. Once the distortion values of the classifiers are set according to the optimization output, the source-channel separation theorem [36] guarantees that (38) will give the optimal rate allocation.

Then, we formulate the following optimization problem that attempts to minimize the mismatch probability for a given transmit rate budget $\mathcal{R}$:

$$\begin{align*}
\text{minimize} & \quad \sum_{t=1}^{T} \beta_t p_t \\
\text{subject to} & \quad \sum_{t=1}^{T} r_t = \sum_{t=1}^{T} (1 - h(p_t)) \leq \mathcal{R} \quad (39) \\
& \quad 0 \leq p_t \leq \frac{1}{2}, \quad t = 1, \ldots, T
\end{align*}$$

with Hamming distortion $p = (p_1, \ldots, p_T)$ (i.e., mismatch probability) for base classifiers. The optimized communication rates $r^* = (r^*_1, \ldots, r^*_T)$ can be readily obtained by (38) since (38) is bijective in the interval $0 \leq p_t \leq \frac{1}{2}$.

**Remark 12:** The rates allocation problem (39) is a convex problem since the binary entropy function $h(p_t)$ is concave.

**Corollary 4:** The optimal solution $p^*$ of (39) satisfies the following condition:

$$\log \frac{1 - p^*_t}{p^*_t} = \frac{\beta_t}{\nu}, \quad (40)$$
TABLE II
SIMULATION SETTINGS

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Transmit power optimization (35)</th>
<th>Transmit rate optimization (39)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Face-MNIST (T-shirt/top vs. Trousers)</td>
<td>CIFAR-10 (Airplane vs. Bird) (Airplane vs. Car)</td>
<td></td>
</tr>
<tr>
<td>Base classifiers</td>
<td>Decision stumps</td>
<td>Neural networks</td>
</tr>
<tr>
<td># of base classifiers</td>
<td>$T = 10, 20$</td>
<td>$T = 10$</td>
</tr>
</tbody>
</table>

where $0 < \beta_t < \infty$ corresponds to $0 < \rho_t^* < \frac{1}{2}$ and $\nu$ is a positive dual variable of the KKT conditions.

**Proof:** The proof is given in Appendix G.

The optimized mismatch probability of $p_t^*$ is achievable if and only if $C_t > 1 - h(p_t^*)$ where $C_t$ denotes the channel capacity between the $t$-th base classifier and the weighted voter. Hence, for Gaussian channels and Rayleigh fading channels, we can obtain the optimized SNRs as in the following examples.

**Example 1 (Gaussian Channels):** For the parallel Gaussian channels, we obtain the optimized transmit power allocation as follows:

$$\text{SNR}_t^* = \left(\frac{\rho_t^*}{\sigma_t^*}\right)^2 = 2^{2\rho_t^*} - 1$$

for $t \in \{1, \ldots, T\}$. The optimized SNR is obtained from $r_t^* = \frac{1}{2}\log_2 (1 + \text{SNR}_t^*)$.

**Example 2 (Rayleigh Fading Channels):** For the Rayleigh fading channel, the capacity depends on the channel state information. If the state information is available at the receiver, the capacity is given by [37, Chap. 14]

$$C = \mathbb{E}\left[\log_2 (1 + \rho \text{SNR})\right] = \frac{1}{\ln 2} \int_0^{\frac{1}{\text{SNR}}} e^{-x} \Gamma(0, \frac{1}{\text{SNR}}) dx$$

where $\rho$ has an exponential density function and $\Gamma(a, z)$ denotes the complementary gamma function, defined by $\Gamma(a, z) = \int_z^{\infty} t^{a-1} e^{-t} dt$ [37]. Since this capacity is an increasing function of SNR, numerical methods such as the bisection method can be used to compute the optimized SNR from $r_t^* = \frac{1}{\ln 2} e^{\frac{1}{\text{SNR}} - 1} \Gamma\left(0, \frac{1}{\text{SNR}}\right)$. Note that $r_t^*$ was obtained by solving (39).

**V. NUMERICAL RESULTS**

We validate the tools and analytic results of Sections III and IV. We compare mismatch probabilities and classification error probabilities of uniform resource allocation and optimized resource allocations for: 1) transmit power optimization and 2) transmit rate optimization.

We present numerical results on two popular datasets: Fashion-MNIST [38] and CIFAR-10 [39]. Since these datasets are originally designed for multiclass classification, we select two classes to form a binary classification problem as in [40]. The noise-free AdaBoost was trained by Algorithm 1 with decision stumps for Fashion MNIST. The boosted neural networks are trained as in [33] for CIFAR-10 where each base classifier is a fully connected neural network. The simulation settings are presented in Table II.

**A. Transmit Power Optimization**

We first present numerical results of transmit power optimization of (35). Fig. 4 and Fig. 5 evaluate the mismatch probabilities and the classification error probabilities by optimizing the transmit power allocation for $T = 10$ and $T = 20$, respectively. We observe that nonuniform power allocations can lower the mismatch probability as well as the classification error probability. Among the three nonuniform power allocations (Markov proxy, Chernoff proxy, and Gaussian proxy), the power allocation based on the Gaussian proxy achieves the best performance. We emphasize that Fig. 4 and Fig. 5 plot the actual mismatch and classification error probabilities over the test set optimized with different proxies, and not the values of the proxies themselves. Note that the horizontal axis corresponds to the total SNR budget $P_c$.

For $T = 10$, the Gaussian proxy allocation improves the SNR over the uniform allocation by $3.7$ dB at $P_c = 0.1$. For $T = 20$, the SNR gain is $4.5$ dB at $P_c = 0.1$. For higher SNR, the mismatch probabilities converge to zero.

The optimization results by the Chernoff proxy are close to the results by the Markov proxy in the low-SNR regime. As the SNR increases, the optimization results by the Chernoff
As the SNR increases, \( \beta \) decreases for low SNR and increases for higher SNR. Hence, \( \beta \) increases for higher SNR. We observe that the Gaussian proxy get close to the results by the Gaussian proxy. It can be explained by the Taylor approximation of \( \beta = e^{s\alpha} - 1 \) in the Chernoff proxy (in Table I) as follows:

\[
e^{s\alpha} - 1 \simeq s\alpha + (s\alpha)^2.
\]

We observe that the optimal \( s^* \) in Algorithm 2 is small for low SNR and increases for higher SNR. Hence, \( \beta \) is approximated to \( s\alpha \) in the low SNR region since \( s\alpha \ll 1 \).

As the SNR increases, \((s\alpha)^2\) becomes a better approximation to \( \beta \). Then, the corresponding coefficients are approximated by \( \beta = s^2(\alpha_1^2, \ldots, \alpha_T^2) \), which is equivalent to the Gaussian proxy optimization.

**B. Transmit Rate Optimization**

We provide numerical results of transmit rate optimization of (39) for the Gaussian channel. Fig. 6 and Fig. 7 evaluate the mismatch probabilities and the classification error probabilities by optimizing the rate allocation for (Airplane vs. Bird) and (Airplane vs. Car), respectively. We observe that the optimized rate allocations can improve the mismatch probability as well as the classification error probability. Among the three nonuniform power allocations (Markov proxy, Chernoff proxy, and Gaussian proxy), the rate optimization based on Gaussian proxy achieves the best performance as in transmit power optimization. Note that the horizontal axis corresponds to the total rate budget \( R \) of (39) and the maximum \( R \) is the same as \( T \) for binary classification. If \( R = T \), then the mismatch probability becomes zero.

The improvement by the optimized transmit rate allocations of Fig. 6 is greater than Fig. 7. It is because the trained \( \alpha \) of (Airplane vs. Car) is closer to the uniform compared to \( \alpha \) of (Airplane vs. Bird) as shown in Fig. 8. This validates that the more nonuniform \( \alpha \) leads to the more improvement by resource allocation optimizations.

**VI. Conclusions**

We addressed the problem of distributed boosting classification over noisy channels and proposed a principled approach to optimize resource allocation. We defined three proxies and the corresponding importance metrics for base classifiers based on the Markov inequality, the Chernoff bound, and the Gaussian approximation. We formulated convex resource-allocation problems to minimize the impact of the channel noise. We showed that the proposed approach can effectively improve the classification accuracy for the additive Gaussian noise model and for coded communication over arbitrary channels. Also, we found that the non-uniform coefficients in boosting offer an advantage over uniform ones (e.g., in bagging) for this noisy inference model. Although we focused...
on AdaBoost, our approach can be extended to any other weighted ensemble algorithms.

We note that the extension to the multiclass problems would be an important future work, building on the fact that several machine learning techniques enable using binary classifiers to solve multiclass problems [7], [41], [42]. Interesting future work includes extending this approach for different channel models and settings, e.g., various wireless channel models and imperfectly characterized mismatch probabilities.

APPENDIX A
PROOF OF THEOREM 1
Suppose that \( f(x_n) = \text{sign}(g(x_n)) > 0 \) where \( g(x_n) = \sum_{t=1}^{T_n} \alpha_t f_t(x_n) = \gamma_n > 0 \). Then,
\[
\sum_{t \in \mathcal{T}_n^+} \alpha_t - \sum_{t \in \mathcal{T}_n^-} \alpha_t = \gamma_n. \tag{44}
\]
where \( \mathcal{T}_n^+ = \{ t \mid f_t(x_n) = 1 \} \) and \( \mathcal{T}_n^- = \{ t \mid f_t(x_n) = -1 \} \), \( \mathcal{T}_n^- \), respectively. By (44) and (21), \( f(x_n) < 0 \) for \( g(x_n) = \gamma_n > 0 \) is equivalent to
\[
\sum_{t \in \mathcal{T}_n^-} \alpha_t \delta_{t,n} - \sum_{t \in \mathcal{T}_n^+} \alpha_t \delta_{t,n} > \frac{\gamma_n}{2}. \tag{45}
\]
By using the Markov’s inequality, the mismatch probability is upper bounded by
\[
P_m(x_n) = \Pr \left( \sum_{t \in \mathcal{T}_n^+} \alpha_t \delta_{t,n} - \sum_{t \in \mathcal{T}_n^-} \alpha_t \delta_{t,n} > \frac{\gamma_n}{2} \right) \leq 2 \cdot \frac{\sum_{t=1}^{T_n} \alpha_t p_t}{\gamma_n}, \tag{46}
\]
where (47) follows from the Markov’s inequality and (48) follows from \( \mathbb{E}[\delta_{t,n}] = p_t \). Similarly, we can obtain the same upper bound for a case of \( f(x_n) < 0 \).

APPENDIX B
PROOF OF THEOREM 2
Suppose that \( f(x_n) > 0 \). Then, the mismatch probability \( p(x_n) \) is given by
\[
P_m(x_n) = \Pr \left( \sum_{t \in \mathcal{T}_n^-} \alpha_t \delta_{t,n} - \sum_{t \in \mathcal{T}_n^+} \alpha_t \delta_{t,n} > \frac{\gamma_n}{2} \right), \tag{49}
\]
which is the same as (46). For any \( s > 0 \),
\[
P_m(x_n) = \Pr \left( e^{s \left( \sum_{t \in \mathcal{T}_n^-} \alpha_t \delta_{t,n} - \sum_{t \in \mathcal{T}_n^+} \alpha_t \delta_{t,n} \right)} > e^{s \frac{\gamma_n}{2}} \right) \leq \mathbb{E} \left[ \exp \left( s \left( \sum_{t \in \mathcal{T}_n^-} \alpha_t \delta_{t,n} - \sum_{t \in \mathcal{T}_n^+} \alpha_t \delta_{t,n} \right) \right) \right], \tag{50}
\]
where (50) comes from the Markov’s inequality.
Note that \( \delta_{t,n} \) for \( t \in \{ 1, \ldots, T \} \) are independent because of independent \( z_t \). Hence,
\[
\mathbb{E} \left[ \exp \left( s \left( \sum_{t \in \mathcal{T}_n^-} \alpha_t \delta_{t,n} - \sum_{t \in \mathcal{T}_n^+} \alpha_t \delta_{t,n} \right) \right) \right] = \prod_{t \in \mathcal{T}_n^-} \mathbb{E} \left[ \exp (s \alpha_t \delta_{t,n}) \right] \cdot \prod_{t \in \mathcal{T}_n^+} \mathbb{E} \left[ \exp (-s \alpha_t \delta_{t,n}) \right]. \tag{51}
\]
In addition,
\[
\mathbb{E} \left[ \exp(s \alpha_t \delta_{t,n}) \right] = (1 - p_t) + p_t \exp(s \alpha_t) = 1 + p_t \{ \exp(s \alpha_t) - 1 \} \leq \exp (p_t (e^{s \alpha_t} - 1)), \tag{52}
\]
where (52) follows from $1 + u \leq \exp(u)$ and $u = p_t (e^{\alpha_t s} - 1)$. Similarly, $\mathbb{E} [\exp(-s \alpha_t \delta_{t,n})] \leq \exp(-p_t (1 - e^{-\alpha_t s})) = \exp(-p_t e^{-\alpha_t s} (e^{\alpha_t s} - 1))$.

By (50), (51), and (52),

$$P_m(x_n) \leq \exp \left( -s \cdot \frac{\gamma_n}{2} + \sum_{t \in T_n^+} p_t (e^{\alpha_t s} - 1) \right)$$

$$- \sum_{t \in T_n^-} p_t e^{-\alpha_t s} (e^{\alpha_t s} - 1)$$

$$\leq \exp \left( -s \cdot \frac{\gamma_n}{2} + \sum_{t \in T_n^+} p_t (e^{\alpha_t s} - 1) \right)$$

$$\leq \exp \left( -s \cdot \frac{\gamma_n}{2} + \sum_{t=1}^T p_t (e^{\alpha_t s} - 1) \right). \quad (53)$$

Similarly, we can obtain the same upper bound for a case of $\sum_{t=1}^T \alpha_t f_t(x_n) = -\gamma_n < 0$.

Finally,

$$P_m = \mathbb{E} [P_m(x_n)]$$

$$\leq \mathbb{E} \left[ \exp \left( -s \cdot \frac{\gamma_n}{2} \right) \right] \cdot \exp \left( \sum_{t=1}^T (e^{\alpha_t s} - 1) \right)$$

$$= \mathbb{E} \left[ \exp \left( -s \cdot \frac{\gamma_n}{2} \right) \right] \cdot (\hat{p}c), \quad (54)$$

where a larger $s$ reduces $\mathbb{E} [\exp(-s \cdot \frac{\gamma_n}{2})]$ while it increases $\exp(\hat{p}c)$.

**APPENDIX C**

**Proof of Theorem 3**

By (22), we showed that $\hat{f}(x_n) = \pm \gamma_n + v_n$. The classification noise term $v_n$ is given by

$$v_n = -2 \left( \sum_{t \in T_n^+} \alpha_t \delta_{t,n} - \sum_{t \in T_n^-} \alpha_t \delta_{t,n} \right). \quad (55)$$

Since we assume that $z_t$s of (7) are independent, $\delta_{t,n}$s are also independent for a given $x_n$. By the central limit theorem, $\sum_{t \in T_n^+} \alpha_t \delta_{t,n}$ and $\sum_{t \in T_n^-} \alpha_t \delta_{t,n}$ can be approximated as Gaussian distributions, respectively. Hence, $v_n$ can be modeled as Gaussian distribution as well.

The mean of $v_n$ is readily derived by using $\mathbb{E}[\delta_{t,n}] = p_t$. The variance of $v_n$ is given by

$$\sigma_n^2 = 4 \sum_{t=1}^T \alpha_t^2 \cdot \mathbb{Var} \[\delta_{t,n}]$$

$$= 4 \sum_{t=1}^T \alpha_t^2 \cdot p_t (1 - p_t), \quad (56)$$

where $\mathbb{Var} \[\delta_{t,n}] = \mathbb{E} \[\delta_{t,n}^2] - \mu_n^2 = p_t (1 - p_t)$.

**APPENDIX D**

**Proof of Theorem 4**

For given $p = (p_1, \ldots, p_T)$ and $\gamma = (\gamma_1, \ldots, \gamma_n)$, the upper bound on $P_m$ is given by

$$h(s) = \left( \frac{1}{N} \sum_{n=1}^N e^{-s \cdot \frac{\gamma_n}{2}} \right) \cdot \exp \left( \sum_{t=1}^T (e^{\alpha_t s} - 1) p_t \right)$$

$$= \frac{1}{N} \cdot h_1(s) \cdot h_2(s), \quad (57)$$

where

$$h_1(s) = \sum_{n=1}^N e^{-s \cdot \frac{\gamma_n}{2}}, \quad (58)$$

$$h_2(s) = \exp \left( \sum_{t=1}^T (e^{\alpha_t s} - 1) p_t \right). \quad (59)$$

We show that $h''(s) \geq 0$ to check the convexity of $h(s)$.

First, we obtain

$$h'(s) = h_1'(s) h_2(s) + h_1(s) h_2'(s)$$

$$= \sum_{n=1}^N \left\{ \left( \sum_{t=1}^T p_t \alpha_t e^{\alpha_t s} - \frac{\gamma_n}{2} \right) e^{-s \cdot \frac{\gamma_n}{2}} \right\} \cdot h_2(s), \quad (60)$$

where $h_2(s) > 0$ for $s > 0$ and $\alpha_t > 0$. Hence, $h'(s) = 0$ can be achieved by $s$ satisfying

$$\sum_{n=1}^N \left\{ \left( \sum_{t=1}^T p_t \alpha_t e^{\alpha_t s} - \frac{\gamma_n}{2} \right) e^{-s \cdot \frac{\gamma_n}{2}} \right\} = 0, \quad (61)$$

which is equivalent to (30) in Algorithm 2.

The second derivative of $h(s)$ is given by

$$h''(s) = h_2(s) \cdot \sum_{n=1}^N e^{-s \cdot \frac{\gamma_n}{2}} \cdot h_3(s), \quad (62)$$

where $h_3(s)$ is given by

$$h_3(s) = \gamma_n^2 - 4 \gamma_n \sum_{t=1}^T p_t \alpha_t e^{\alpha_t s}$$

$$+ 4 \left\{ \sum_{t=1}^T p_t \alpha_t e^{\alpha_t s} \cdot \left( \sum_{t=1}^T p_t \alpha_t e^{\alpha_t s} \right) \right\}$$

$$= \left( \gamma_n - 2 \gamma_n \sum_{t=1}^T p_t \alpha_t e^{\alpha_t s} \right)^2 + 4 \sum_{t=1}^T p_t \alpha_t^2 e^{\alpha_t s}. \quad (63)$$

Note that $h_2(s) > 0$ and $h_3(s) > 0$ for $s > 0$, $\alpha_t \geq 0$ and $p_t \geq 0$ for any $t$. Hence, $h''(s) > 0$ and the optimal $s$ should satisfy (61).

**APPENDIX E**

**Proof of Theorem 5**

The Lagrangian $L(s, \nu, \lambda)$ of (35) is given by

$$L(s, \nu, \lambda)$$

$$= \sum_{t=1}^T \beta_t Q \left( \frac{s_t}{\alpha_t} \right) + \nu \left( \sum_{t=1}^T s_t^2 - C \right) - \sum_{t=1}^T \lambda_t s_t. \quad (64)$$
The corresponding KKT conditions are as follows:

\[
\sum_{t=1}^{T} s_t^2 \leq C, \quad \nu \geq 0, \quad \nu \cdot \left( \sum_{t=1}^{T} s_t^2 - C \right) = 0, \quad (65)
\]
\[
s_t \geq 0, \quad \lambda_t \geq 0, \quad \lambda_t s_t = 0, \quad (66)
\]
\[
\frac{\partial L}{\partial s_t} = -\frac{\beta_t}{\sqrt{2\pi}\sigma_t} \exp \left( -\frac{s_t^2}{2\sigma_t^2} \right) + 2\nu s_t - \lambda_t = 0 \quad (67)
\]
for \( t \in \{1, \ldots, T\} \).

From (67), \( \lambda_t = 2\nu s_t - \frac{\beta_t}{\sqrt{2\pi}\sigma_t} \exp \left( -\frac{s_t^2}{2\sigma_t^2} \right) \). (68)

If \( \nu = 0 \), then \( \lambda_t < 0 \), which violates (66) because of \( \beta_t > 0 \). Hence, we claim that \( \nu \neq 0 \), which results in \( \sum_{t=1}^{T} s_t^2 = C \).

If \( s_t = 0 \), then \( \lambda_t = -\frac{\beta_t}{\sqrt{2\pi}\sigma_t} \), which violates (66). Hence, we claim that \( s_t > 0 \) and \( \lambda_t = 0 \), i.e.,

\[
s_t = \frac{\beta_t}{2\sqrt{2\pi}\sigma_t^2} \exp \left( -\frac{s_t^2}{2\sigma_t^2} \right), \quad (69)
\]

which is equivalent to

\[
ts_t^2 \sigma_t^2 \exp \left( \frac{s_t^2}{\sigma_t^2} \right) = \frac{\beta_t^2}{8\pi^2\sigma_t^4}, \quad (70)
\]

By setting \( x = \frac{s_t^2}{\sigma_t^2} \), we obtain \( x \exp(x) = \frac{\beta_t^2}{8\pi^2\sigma_t^4} \). Hence,

\[
x = \text{SNR}_t = W \left( \frac{\beta_t^2}{8\pi^2\sigma_t^4} \right). \quad \text{(APPENDIX F)}
\]

**PROOF OF THEOREM 3**

By replacing \( Q(x) \) with its Chernoff bound \( \frac{1}{2} \exp \left( -\frac{x^2}{2} \right) \), the optimization problem (35) will be modified to

\[
\text{minimize} \quad \sum_{t=1}^{T} \frac{\beta_t}{2} \exp \left( -\frac{x_t^2}{2} \right)
\]
\[
\text{subject to} \quad \sum_{t=1}^{T} x_t^2 \leq \frac{C}{\sigma^2}, \quad x_t \geq 0 \quad \text{for} \quad t = 1, \ldots, T, \quad (71)
\]

where \( x_t = \frac{x_t}{\sigma_t} \). The corresponding Lagrangian is given by

\[
L(x, \nu, \lambda) = \sum_{t=1}^{T} \frac{\beta_t}{2} \exp \left( -\frac{x_t^2}{2} \right) + \nu \left( \sum_{t=1}^{T} x_t^2 - C \right) - \sum_{t=1}^{T} \lambda_t x_t. \quad (72)
\]

From the KKT conditions, we can obtain the following conditions of the optimal \( x^* \) (as in Appendix E):

\[
\sum_{t=1}^{T} (x_t^*)^2 = \frac{C}{\sigma^2}. \quad (73)
\]
\[
\frac{\beta_t}{2} \exp \left( -\frac{(x_t^*)^2}{2} \right) = 2\nu. \quad (74)
\]

By (71) and (74), the optimized proxy is given by

\[
\sum_{t=1}^{T} \frac{\beta_t}{2} \exp \left( -\frac{(x_t^*)^2}{2} \right) = 2T\nu. \quad (75)
\]

Note that (74) is equivalent to

\[
(x_t^*)^2 = \text{SNR}_t^* = 2\log \frac{\beta_t}{4\nu}. \quad (76)
\]

By (73) and (76),

\[
\log \prod_{t=1}^{T} \frac{\beta_t}{4\nu} = \frac{C}{2\sigma^2}, \quad (77)
\]

which leads to \( \nu = \frac{1}{2} \exp \left( -\frac{C}{2T\sigma^2} \right) \left( \prod_{t=1}^{T} \beta_t \right)^{\frac{1}{2}} \). By (75), the optimized proxy is given by

\[
2T\nu = \frac{T}{2} \exp \left( -\frac{C}{2T\sigma^2} \right) \left( \prod_{t=1}^{T} \beta_t \right)^{\frac{1}{2}}. \quad (78)
\]

**APPENDIX G**

**PROOF OF COROLLARY 4**

The Lagrangian \( L(p, \nu, \lambda, \eta) \) of (39) is given by

\[
L(p, \nu, \lambda, \eta) = \sum_{t=1}^{T} \beta_t p_t + \nu \left( \sum_{t=1}^{T} (1 - h(p_t)) - \mathcal{R} \right) - \sum_{t=1}^{T} \lambda_t p_t + \sum_{t=1}^{T} \eta_t \left( p_t - \frac{1}{2} \right). \quad (79)
\]

The corresponding KKT conditions are as follows:

\[
\sum_{t=1}^{T} (1 - h(p_t)) \leq \mathcal{R}, \quad \nu \geq 0, \quad \nu \cdot \left( \sum_{t=1}^{T} (1 - h(p_t)) - \mathcal{R} \right) = 0, \quad (80)
\]
\[
0 \leq p_t \leq \frac{1}{2}, \quad \lambda_t p_t = 0, \quad \lambda_t \geq 0, \quad (81)
\]
\[
\eta_t \left( p_t - \frac{1}{2} \right) = 0, \quad \eta_t \geq 0, \quad (82)
\]
\[
\frac{\partial L}{\partial p_t} = \beta_t - \nu \log \frac{1 - p_t}{p_t} - \lambda_t + \eta_t = 0 \quad (83)
\]

for \( t \in \{1, \ldots, T\} \).

From (84), we obtain

\[
\nu \log \frac{1 - p_t}{p_t} = \beta_t - \lambda_t + \eta_t. \quad (85)
\]

If \( \nu = 0 \), then \( \beta_t = \lambda_t - \eta_t \). Since \( \beta_t > 0 \), we set \( \lambda_t > \eta_t \geq 0 \), which leads to a trivial solution of \( p_t = 0 \) for all \( t \in \{1, \ldots, T\} \) due to complementary slackness of (82). Hence, we claim that \( \nu \neq 0 \) and \( \sum_{t=1}^{T} (1 - h(p_t)) = \mathcal{R} \).

Next, we consider three cases of i) \( 0 < p_t < \frac{1}{2} \), ii) \( p_t = 0 \), and iii) \( \frac{1}{2} < p_t < 1 \): Due to complementary slackness of (83) and (84), we obtain \( \lambda_t = \eta_t = 0 \). Hence, \( \log \frac{1 - p_t}{p_t} = \beta_t \), which is the same as (40).

ii) \( p_t = 0 \): By (83) and (85), we obtain \( \eta_t = 0 \). Then, the RHS of (85) becomes \( \beta_t - \lambda_t \leq \beta_t \) whereas the LHS of (85) goes to infinity. Hence, \( p_t = 0 \) cannot be a solution.

iii) \( p_t = \frac{1}{2} \): By (82) and (85), we obtain \( \lambda_t = 0 \). Then, the RHS of (85) becomes \( \beta_t + \eta_t \) whereas the LHS of (85)
becomes zero. Then, $\beta_t = 0$ and $y_t = 0$. Since $\beta_t > 0$, $p_t = \frac{1}{2}$ cannot be a solution.

Hence, we conclude that the optimal $p_t^*$ satisfies (40).

REFERENCES


Junyoung Shin is an undergraduate student at the Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu, South Korea. He was a research intern in the Department of Electrical Engineering and Computer Science at DGIST. His research interests include machine learning, information theory, and natural language processing.

Yuval Cassuto (S’02–M’08–SM’14) is a faculty member at the Viterbi Department of Electrical and Computer Engineering, Technion — Israel Institute of Technology. His research interests lie at the intersection of the theoretical information sciences and the engineering of practical computing and storage systems. He has served on the technical program committees of leading conferences in both theory and systems. During 2010–2011 he has been a Scientist at EPFL, the Swiss Federal Institute of Technology in Lausanne. From 2008 to 2010 he was a Research Staff Member at Hitachi Global Storage Technologies, San Jose Research Center. In 2018–2019 he held a Visiting Professor position at Western Digital Research, and a Visiting Scholar position at UC Berkeley. He received the B.Sc degree in Electrical Engineering, summa cum laude, from the Technion in 2001, and the M.S. and Ph.D. degrees in Electrical Engineering from the California Institute of Technology, in 2004 and 2008, respectively. From 2000 to 2002, he was with Qualcomm, Israel R&D Center, where he worked on modeling, design and analysis in wireless communications. Dr. Cassuto has won the Best Student Paper Award in data storage from the IEEE Communications Society in 2010 as a student, and in 2019 as an adviser. He also won faculty awards from Qualcomm, Intel, and Western Digital. Before that, he won the 2001 Texas Instruments DSP and Analog Challenge $100,000 prize.

Lav R. Varshney (S’00–M’10–SM’15) received the B.S. degree (magna cum laude) in electrical and computer engineering with honors from Cornell University, Ithaca, New York, in 2004. He received the S.M., E.E., and Ph.D. degrees, all in electrical engineering and computer science, from the Massachusetts Institute of Technology, Cambridge, in 2006, 2008, and 2010, where his theses received the E. A. Guillemin Thesis Award and the J.-A. Kong Award Honorable Mention. He is an associate professor in the Department of Electrical and Computer Engineering and the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, with further appointments in computer science, industrial engineering, neuroscience, digital agriculture, and precision nutrition. He is also affiliated with the Discovery Partners Institute of the University of Illinois System. He is currently a White House Fellow, serving in the National Security Council, Cyber and Emerging Technology, Washington, DC, USA. During 2019–2020, he was a principal research scientist at Salesforce Research, Palo Alto, CA, USA. During 2010–2013, he was a research staff member at the IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA. His research interests include information theory and artificial intelligence.