2021 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, OCT. 25–28, 2021, GOLD COAST, AUSTRALIA

HYPHYLEARN: A DOMAIN ADAPTATION-INSPIRED APPROACH TO CLASSIFICATION USING LIMITED NUMBER OF TRAINING SAMPLES

Alireza Nooraiepour, Waheed U. Bajwa, and Narayan B. Mandayam

WINLAB, Department of Electrical and Computer Engineering, Rutgers University, NJ, USA

ABSTRACT

The fundamental task of classification given a limited number of training data samples is considered for physical systems with known parametric statistical models. As a solution, a hybrid classification method—termed HYPHYLEARN—is proposed that exploits both the physics-based statistical models and the learning-based classifiers. The proposed solution is based on the conjecture that HYPHYLEARN would alleviate the challenges associated with the individual approaches of learning-based and statistical model-based classifiers by fusing their respective strengths. The proposed hybrid approach first estimates the unobservable model parameters using the available (suboptimal) statistical estimation procedures, and subsequently uses the physics-based statistical models to generate *synthetic data*. Next, the training data samples are incorporated with the synthetic data in a learning-based classifier that is based on domain-adversarial training of neural networks. Numerical results on multiuser detection, a concrete communication problem, demonstrate that HYPHYLEARN leads to major classification improvements compared to the existing stand-alone and hybrid classification methods.

1. INTRODUCTION

We revisit the problem of classification with limited number of training data samples in this paper. The fundamental task of classification comes up in various fields and is traditionally tackled within two frameworks: 1) statistical setting, and 2) fully data-driven setting. In the first case, the classification problem is usually dealt with within a hypothesis testing (HT) framework based on the assumption that data generation adheres to a known probabilistic model. However, these models might rely on a large number of unobservable parameters, estimation of which from limited number of data samples could be a major hurdle. The fully data-driven (i.e., learning based) setting, on the other hand, relies on a large number of data samples for finding an optimal mapping from the data samples to the corresponding labels. But availability of such data in many application scenarios is generally limited, which might lead to learning of a suboptimal map.

This work was supported by the National Science Foundation (NSF) under grants OAC-1940074 and ECCS-2028823.

The overarching objective of this paper is to develop an algorithmic framework for classification from limited number of training data samples in applications in which neither model-based nor learning-based approaches alone result in very good classification performance. Our goal in this context is to develop a classification framework that can deal with the difficulties associated with stand-alone methods via a hybrid approach that consolidates physics-based and fully data-driven classification approaches. There have been previous attempts to incorporate physics-inferred information in the fully data-driven setting. In the field of wireless communications, for instance, the authors in [1] employ deep transfer learning (DTL) to solve a specific resource management problem. Specifically, they utilize abundant data from an approximate resource allocation model along with limited data from the unknown physical model in the DTL fine-tuning approach [2, 3]. Closer to the idea of physics-guided machine learning, a recurrent neural network (RNN) is modified in [4] to incorporate information from the physics-based model as an internal state of the RNN. Furthermore, parameters of the physics-based models are combined with sensor readings and used as inputs to a DNN to develop a hybrid prognostics model in [5]. However, such works do not consider the difficulties associated with estimating the model parameters, which would lead to inaccurate physics-based statistical models. The resulting discrepancy between the model and the underlying physical process asks for a learning-based classifier that is capable of leveraging the data in a way to alleviate this mismatch problem.

The physics-based classification approach, despite the potential difficulty in properly estimating the model parameters, retains essential prior information about the system's behavior. At the same time, a learning-based classifier is a powerful tool for finding patterns and discriminative representations from a given dataset. However, the paucity of data poses the main challenge towards devising a stand-alone classification approach in both cases. In this vein, we focus on the task of classification for a physical process assuming that a limited number of training data samples is available. We consider the case where the physical process can be described by physics-based parametric statistical models. As these models tend to be complex in general, estimation of the unknown model parameters using the maximum likelihood estimation (MLE)

978-1-7281-6338-3/21/$31.00 ©2021 IEEE

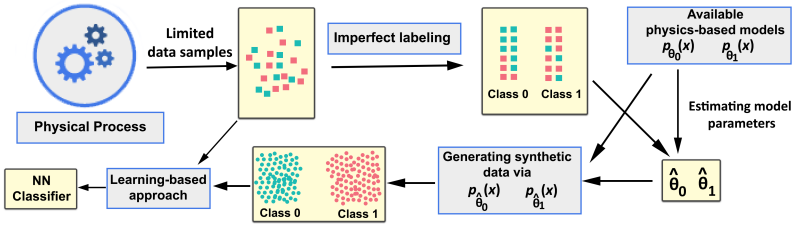

Fig. 1: A schematic of our proposed hybrid classification approach (HYPHYLEARN) illustrated for a binary classification setting, which exploits both physics-based statistical models and learning-based classifiers.

procedure could be computationally prohibitive.

We propose HYPHYLEARN—a novel hybrid classification method—as a solution, which exploits both physics-based statistical models and learning-based classifiers. This approach makes use of (necessarily suboptimal) parameter estimation algorithms to obtain (approximate) parameter estimates. Next, plugging in these estimates in the physics-based statistical models enables us to generate *synthetic* data. HYPHYLEARN then relies on neural networks (NNs), which are powerful tools for finding a discriminative feature space, towards obtaining a learning-based classifier. Specifically, the learning process involves training a NN to map the training and synthetic data to a common space under which they are indistinguishable. In the meantime, a learning-based classifier is trained on the mapping of the synthetic data in the new space aiming to find discriminative features. Indeed, learning such common feature space addresses the distribution mismatch problem between the training data samples and the generated synthetic data due to the errors in parameter estimation. It is then expected that the classifier trained on the mapped synthetic data will perform better on both data distributions. The overall learning process to alleviate the mismatch problem results in the domain-adversarial training [6] of the NNs. A schematic of HYPHYLEARN for a binary classification example is illustrated in Fig. 1.

The rest of the paper is organized as follows. The problem is formally posed in Section 2. Our proposed solution is described in Section 3, which discusses various pieces of the HYPHYLEARN approach. We introduce the case study, which concerns the multiuser detection problem (MUD), in Section 4, and present numerical results regarding the application of HYPHYLEARN and other existing methods for MUD in Section 5. Finally, the paper is concluded in Section 6.

2. PROBLEM FORMULATION

Consider a physical process consisting of C distinct behaviors where the physics-based parametric statistical model for the ith behavior is available in the form of a parametric probability density function (PDF) denoted by the conditional prior $p_i(\mathbf{x};\boldsymbol{\theta}_i)$ on observations $\mathbf{x}$ that belong to an observation space $\mathcal{X}$. Assuming the true underlying parameter for the ith behavior is $\boldsymbol{\theta}_i^*$, the data for this behavior is generated by drawing independent and identically distributed (i.i.d.) samples from $p_i(\mathbf{x};\boldsymbol{\theta}_i^*)$. Assuming further that the ith behavior is chosen with a prior probability π_i, our goal is to devise a decision rule to determine a given sample $\mathbf{x} = [x_1, \ldots, x_n]^T$ is generated under which behavior. Clearly, this can be cast as a C-ary classification problem via $H_i : \mathbf{x} \sim p_i(\mathbf{x};\boldsymbol{\theta}_i^*),\ i = 0, \ldots, C-1$. We consider the case where this decision is made by a classifier $h_{\boldsymbol{\phi}}(\cdot)$ parameterized by $\boldsymbol{\phi} \in \mathbb{R}^d$, $h_{\boldsymbol{\phi}}(\mathbf{x}) : \mathcal{X} \to \{0, \ldots, C-1\}$, which partitions $\mathcal{X}$ into C disjoint sets, $\{\mathcal{X}_i\}$, and decides in favor of H_i if $\mathbf{x} \in \mathcal{X}_i$. We note that the optimal classifier in this setting is given by the Bayes decision rule, i.e., $h_{\phi^*}(\mathbf{x}) = \text{argmax}_{i=0,\ldots,C-1}\ \pi_i p_i(\mathbf{x};\boldsymbol{\theta}_i^*)$ [7] which relies on the true values of the parameters, i.e., $\boldsymbol{\theta}_i^*$'s.

We focus on the case where although the parametric model $p_i(\mathbf{x};\boldsymbol{\theta}_i)$ is known for the ith behavior, one does not have access to the corresponding underlying true parameter $\boldsymbol{\theta}_i^*$. Instead, only a small number of training data generated in an i.i.d. manner from $p_i(\mathbf{x};\boldsymbol{\theta}_i^*), \forall i$, are available. Specifically, we denote the available dataset by $\mathcal{D}_r = \{\mathbf{x}_{r,n}\}_{n=1}^{N_r}$, where N_r is the total number of data samples. Also, the corresponding ground-truth label for the nth sample is denoted by $y_{r,n}$, which is only given for $N_{r,l}$ number of data samples where $N_{r,l} \leq N_r$. Furthermore, we consider the case where the model $p_i(\mathbf{x};\boldsymbol{\theta}_i)$ under the ith behavior is a non-trivial function of the underlying parameter for which conventional estimation procedures such as maximum likelihood estimation (MLE) are computationally prohibitive to implement. The implication of this aspect of the problem formulation is that the performance of any suboptimal parameter estimation method is bound to be limited. As a result, statistical model-based methods which plug-in these estimates in $p_i(\mathbf{x};\boldsymbol{\theta}_i)$, i.e., *plug-in classifiers*, would have a deteriorated performance as well.

Unlike the classifiers that rely heavily on the knowledge of the parametric statistical models and the estimated parameters, a purely data-driven approach can result in a classifier that disregards the available parametric models. However, as the data generation processes are governed by non-trivial models, a large number of data is needed in this case to extract related patterns from each behavior that can lead to a highly discriminative feature space. By noting that the performances of the fully data-driven and the statistical model-based classifiers are particularly curbed when they are used in a stand-alone fashion, we conjecture that fusing the strengths of the two can

lead to a superior classification algorithm in our setting, as described in the next section.

3. PROPOSED SOLUTION: HYPHYLEARN

The main deciding factor in superiority of a solution for the problem setup introduced in Section 2 is the extent to which it exploits the available information, i.e., training data and the parametric statistical models ($p_i(\mathbf{x};\boldsymbol{\theta}_i)$). In particular, the plug-in classifiers tend not to exploit this information in the most optimal fashion as performance of practical parameter estimation procedures might be limited. We propose a novel hybrid classification method to make use of the available information in learning-based classifiers, which are powerful tools for finding discriminative feature spaces. In the following, we describe the various steps of the proposed solution that is termed HYPHYLEARN in detail.

Step 1—Imperfect labeling: As the available data are not assumed to be completely labeled in our problem setup, the first step in our solution deals with assigning labels to the unlabeled data samples in $\mathcal{D}_r$. This involves a clustering step that partitions the dataset $\mathcal{D}_r$ into C distinct groups. Then, the groups are labeled using the available $N_{r,l}$ labels. For example, a label can be assigned to a group based on the number of labeled training data it includes from each behavior. If the majority of such samples corresponds to the ith behavior, the group is labeled as i. Subsequently, we refer to a group assigned with the label i by $\mathcal{D}_{r,i}$ for $i = 0, \ldots, C-1$. We denote this imperfect labeling process by $g(\mathbf{x}) : \mathcal{X} \rightarrow \{0, \ldots, C-1\}$ in the remainder of the paper. We also refer to the number of samples in the cluster labeled as i by $N_{r,i}$.

Step 2—Parameter estimation: Based on the labels assigned in Step 1 to the unlabeled data samples, we estimate the parameters of the physics-based statistical models under each behavior. To this end, we utilize $\mathcal{D}_{r,i}$ to estimate the parameter vector $\boldsymbol{\theta}_i^*$ corresponding to the ith behavior. Furthermore, the priors are estimated as $\widehat{\pi}_i = N_{r,i}/N_r$. We recall from our problem setup that the MLE, which is usually utilized for parameter estimation purposes, cannot be employed here due to the formidable complexity of $p_i(\mathbf{x};\boldsymbol{\theta}_i)$. Instead, a (necessarily) suboptimal method, $T(\cdot)$, is utilized to estimate the parameters as $\widehat{\boldsymbol{\theta}}_i = T(\mathcal{D}_{r,i})$ for all the behaviors. The parameter estimation performance is limited here due to both the suboptimality of $T(\cdot)$ and mislabeled samples in $\mathcal{D}_{r,i}, \forall i$.

Step 3—Forming a synthetic dataset: The paucity of available data in our problem formulation seems to preclude utilization of a learning-based classifier as part of the solution. However, we note that the available physics-based statistical models, in the form of parametric PDFs, enable us to generate synthetic data to augment the available data, and make it possible to exploit the discriminative power of learning-based classifiers. Having access to the estimated parameter $\widehat{\boldsymbol{\theta}}_i$ obtained in Step 2, we plug it in the available physics-based statistical model to obtain a PDF $p_i(\mathbf{x};\widehat{\boldsymbol{\theta}}_i)$ for the ith behavior. In order to generate a synthetic dataset, we first sample w from a categorical distribution parameterized by $\widehat{\boldsymbol{\pi}} = [\widehat{\pi}_0, \ldots, \widehat{\pi}_{C-1}]$ over the sample space of $\{0, \ldots, C-1\}$. Then, we sample a data point $\mathbf{x}_{s,i}$ according to $\mathbf{x}_{s,i} \sim p_w(\mathbf{x};\widehat{\boldsymbol{\theta}}_w)$ with the associated label $y_{s,i} = w$. Repeating this process N_s number of times, we obtain a synthetic dataset $\mathcal{D}_s = \{\mathbf{x}_{s,i}, y_{s,i}\}_{i=1}^{N_s}$ in which $\mathbf{x}_{s,i}$'s are statistically independent.

Step 4—Incorporating synthetic and training data in a learning-based classifier: The errors introduced during the labeling and the parameter estimation steps that precede the synthetic data generation process incur a mismatch between the distributions corresponding to the training and synthetic datasets. The question is how a learning-based classifier can be trained to alleviate this problem. For example, in the fine-tuning approach [1], a NN-based classifier will be trained on the synthetic data first, and then training data are used to refine the weights of the corresponding NN. However, we conjecture that such learning strategies that utilize the training and synthetic data in the separate stages of training are not the best solution here; rather, synthetic and training data should jointly be incorporated in a learning-based classifier. To this end, inspired by the domain-adversarial training of the neural networks [6] in the domain-adaptation literature, we propose to map the synthetic and training data through a (deep) NN $M_{\boldsymbol{\psi}} : \mathcal{X} \rightarrow \mathcal{Z}$, which is parameterized by a real vector $\boldsymbol{\psi}$, into a common feature space $\mathcal{Z}$. Consequently, a (deep) NN-based classifier $h_{\boldsymbol{\phi}_1}(\mathbf{z})$, parameterized by $\boldsymbol{\phi}_1$, which is trained on the synthetic data within the space $\mathcal{Z}$ is expected to perform well on both the training and synthetic data. Similar to [6, 8] a third NN, $d_{\boldsymbol{\zeta}}$ parameterized by $\boldsymbol{\zeta}$, is also utilized aiming at classifying between the mapped training and synthetic data samples.

Specifically, we assume the input and output layers of the NNs corresponding to $M_{\boldsymbol{\psi}}$ have n_x and n_z number of neurons, respectively, which denote the dimensions of the spaces $\mathcal{X}$ and $\mathcal{Z}$, respectively. Subsequently, the input layer of $h_{\boldsymbol{\phi}_1}$ has n_z neurons while its output layer contains C neurons whose activation function is chosen to be the softmax function $\boldsymbol{\sigma}(\mathbf{z})$ for which the ith element is given by $\frac{e^{\mathbf{z}[i]}}{\sum_{i=1}^{n_z} e^{\mathbf{z}[i]}}$. Note that $\mathbf{z}[i]$ denotes the ith element of the vector $\mathbf{z}$. Consequently, the classification error associated with $h_{\boldsymbol{\phi}_1}$ over the synthetic dataset $\mathcal{D}_s$ equals the averaged cross-entropy loss, i.e.,

$$\mathcal{L}_s(\boldsymbol{\psi}, \boldsymbol{\phi}_1 | \mathcal{D}_s) = \frac{1}{n_s} \sum_{n=1}^{n_s} \sum_{i=1}^{C} \mathbf{l}_{s,n}[n] \log \mathbf{y}_{\boldsymbol{\psi},\boldsymbol{\phi}_1,\mathbf{x}_{s,n}}[n], \quad (1)$$

where $\mathbf{l}_{s,n} = \mathbf{e}_C(y_{s,n})$ denotes the one-hot encoded version [1] of the label $y_{s,n}$ corresponding to the nth sample. Regarding $d_{\boldsymbol{\zeta}}$, we consider a NN with n_z input neurons and 2 output neurons with softmax activation function. This classifier is

[1] The vector $\mathbf{e}_n(y)$, *one-hot* encoded version of a non-negative integer y, equals to an all-zero vector of length n except for its yth element which is set to 1.

Algorithm 1: HYPHYLEARN

Input: Parametric models $p_i(\mathbf{x};\boldsymbol{\theta}_i)$ $(i=0,\ldots,C-1)$; Training dataset $\mathcal{D}_r=\{\mathbf{x}_{r,n}\}_{n=1}^{N_r}$; learning rates μ_{r_1}, μ_{r_2}, μ_{r_3}; Mini-batch size $N_b<N_r$; Number of synthetic data samples N_s

Output: The mapping $M_{\boldsymbol{\psi}}(\cdot)$ and the classifier $h_{\boldsymbol{\phi}_1}(\cdot)$, parameterized by the real vectors $\boldsymbol{\psi}$ and $\boldsymbol{\phi}_1$, respectively

// Step 1 - Imperfect labeling

$\{\mathcal{D}_{r,0},\ldots,\mathcal{D}_{r,C-1}\}\leftarrow$ Applying $g(\mathbf{x})$ to unlabeled samples

// Step 2 - Parameter estimation

$\widehat{\boldsymbol{\theta}}_j\leftarrow T_j(\mathcal{D}_{r,j})$, $\widehat{\pi}_j\leftarrow\frac{|\mathcal{D}_{r,j}|}{N_r}$ for $j=0,\ldots,C-1$

// Step 3 - Forming a synthetic dataset

$p_i(\mathbf{x};\widehat{\boldsymbol{\theta}}_i)\leftarrow$Plug $\widehat{\boldsymbol{\theta}}_i$ in $p_i(\mathbf{x};\theta_i)$ for $i=0,\ldots,C-1$

for $n=1$ ***to*** N_s **do**

  $r\sim$ unif$(0,1)$, $w=\text{argmin}_k\sum_{i=0}^{k-1}\widehat{\pi}_i\geq r$

  $\mathbf{x}_{s,n}\sim p_w(\mathbf{x};\widehat{\boldsymbol{\theta}}_w)$, $y_{s,n}=w$

  Add $\{\mathbf{x}_{s,n},y_{s,n}\}$ to $\mathcal{D}_s$

end

// Step 4 - Learning-based classifier

repeat

  $\mathcal{D}_{r,b}\leftarrow N_b$ random samples from $\mathcal{D}_r$, $\mathcal{D}_{s,b}\leftarrow N_b$ random samples from $\mathcal{D}_s$

  // Forward propagation via (1), (3)

  $L_s\leftarrow\mathcal{L}_s(\boldsymbol{\psi},\boldsymbol{\phi}_1|\mathcal{D}_{s,b})$

  $L_c\leftarrow\mathcal{L}_c(\boldsymbol{\psi},\boldsymbol{\zeta}|\mathcal{D}_{r,b},\mathcal{D}_{s,b})$

  // Backward propagation

  $\mathcal{G}_{s,\boldsymbol{\phi}_1}\leftarrow\nabla_{\boldsymbol{\phi}_1}L_s$, $\mathcal{G}_{s,\boldsymbol{\psi}}\leftarrow\nabla_{\boldsymbol{\psi}}L_s$

  $\mathcal{G}_{c,\boldsymbol{\zeta}}\leftarrow\nabla_{\boldsymbol{\zeta}}L_c$, $\mathcal{G}_{c,\boldsymbol{\psi}}\leftarrow\nabla_{\boldsymbol{\psi}}L_c$

  $\boldsymbol{\psi}\leftarrow\boldsymbol{\psi}-\mu_{r_1}(\mathcal{G}_{s,\boldsymbol{\psi}}-\mathcal{G}_{c,\boldsymbol{\psi}})$, $\boldsymbol{\phi}_1\leftarrow\boldsymbol{\phi}_1-\mu_{r_2}\mathcal{G}_{s,\boldsymbol{\phi}_1}$, $\boldsymbol{\zeta}\leftarrow\boldsymbol{\zeta}-\mu_{r_3}\mathcal{G}_{c,\boldsymbol{\zeta}}$

until convergence;

trained to distinguish between the training and synthetic data in the $\mathcal{Z}$ space labeled as 0 and 1, respectively. Consequently, by defining a two-dimensional vector $d_{\boldsymbol{\psi},\boldsymbol{\zeta},\mathbf{x}}\stackrel{\triangle}{=}d_{\boldsymbol{\zeta}}\big(M_{\boldsymbol{\psi}}(\mathbf{x})\big)$, the $\widehat{d}_{\mathcal{A}_\Phi}$ term can be approximated by the cross-entropy loss associated with $d_{\boldsymbol{\zeta}}$ as follows:

$$\mathcal{L}_d(\boldsymbol{\psi},\boldsymbol{\zeta}|\mathcal{D}_s,\mathcal{D}_r)=2\big(1-2\mathcal{L}_c(\boldsymbol{\psi},\boldsymbol{\zeta}|\mathcal{D}_s,\mathcal{D}_r)\big),\quad(2)$$

$$\mathcal{L}_c(\boldsymbol{\psi},\boldsymbol{\zeta}|\mathcal{D}_s,\mathcal{D}_r)=\frac{1}{n_r}\sum_{i=1}^{n_r}\log d_{\boldsymbol{\psi},\boldsymbol{\zeta},\mathbf{x}_{r,n}}[1]+\frac{1}{n_s}\sum_{n=1}^{n_s}\log d_{\boldsymbol{\psi},\boldsymbol{\zeta},\mathbf{x}_{s,n}}[2].\quad(3)$$

For the joint learning of the desired feature map and the classifier, the NNs $M_{\boldsymbol{\psi}}$ and $h_{\boldsymbol{\phi}_1}$ should be trained to minimize the sum of the losses in (1) and (2), while the classifier $d_{\boldsymbol{\zeta}}$ is trained to minimize (3). In particular, the saddle points $\widehat{\boldsymbol{\psi}}$, $\widehat{\boldsymbol{\phi}}_1$ and $\widehat{\boldsymbol{\zeta}}$ can be found via the stochastic gradient descent algorithm, which leads, to the adversarial training of the above three NNs [6]. As an algorithmic framework, HYPHYLEARN is described in Algorithm 1.

4. CASE STUDY: MULTIUSER DETECTION

An important problem in multipoint-to-point digital communication networks (e.g., radio networks) is the optimum centralized demodulation of the information sent simultaneously by several users through a Gaussian multiple-access channel (MAC). Even though the users may not employ a protocol to coordinate their transmission epochs, effective sharing of the channel is possible because each user modulates a different signature signal waveform. In this section, we consider the uplink of a cellular communication system where K users are asynchronously sharing a channel to communicate with a base station (BS). The problem of multiuser detection (MUD) in this setting amounts to inferring the information bit associated with each user from the received signals from the MAC.

At the BS, a discrete model for the baseband equivalent of the received signal from the Kth user can be obtained by relying on the notion of effective chip pulse, denoted by $g_k(t,\tau_k)$ for the kth user experiencing a timing offset of τ_k. By definition, we have $g_k(t,\tau_k)=A_k h_{RC}(t-\tau_k)^* c_k(t)$ where A_k denotes the complex amplitude of the kth user, $h_{RC}(t)$ represents a raised cosine chip waveform time-limited to $[0,8Tc)$, and $c_k(t)$ is the impulse response modeling the channel effects between the BS and the kth user. We assume the channel impulse response (CIR), $c_k(t)$, takes the form of a time-invariant multipath channel with L paths, i.e., $c_k(t)=\sum_{l=0}^{L-1}\alpha_{k,l}\delta(t-\tau'_{k,l})$, which is parameterized by the complex path gains $\alpha_{k,l}$ and the corresponding path delays $\tau'_{k,l}$. The kth user employs a pseudo-noise (PN) code $\{\beta_{k,p}^{(n)}\}_{n=0}^{N-1}$ for spreading its data bit $b_k(p)$ on the pth symbol interval, where N is the processing gain. In fact, N defines the ratio between the chip interval T_c and bit interval duration, i.e., $T_c=T_b/N$. By sampling the received signal at a rate M/T_c, one obtains $\mathbf{g}_k\in\mathbb{C}^{MN+8M-1\times 1}$ as $\mathbf{g}_k=\big[g_k(T_c/M,\tau_k),g_k(2T_c/M,\tau_k),\ldots,g_k(T_b+(8M-1)T_c/M,\tau_k)\big]^T$. Using $\mathbf{g}_k$ one can obtain a compact model for the received samples as a MN-dimensional vector, $\mathbf{y}(p)$, in the pth symbol interval $\mathcal{I}_p=[pT_b,(p+1)T_b]$ such that [9]

$$\mathbf{y}(p)=\sum_{k=0}^{K-1}\mathbf{A}_k(p)\mathbf{g}_k+\mathbf{n}(p)=\mathbf{A}(p)\mathbf{g}+\mathbf{n}(p),\quad(4)$$

for $\mathbf{A}(p)=[\mathbf{A}_0(p),\ldots,\mathbf{A}_{K-1}(p)]$ and $\mathbf{A}_k(p)=b_k(p-2)\mathbf{C}_{k,p-2}(p)+b_k(p-1)\mathbf{C}_{k,p-1}(p)+b_k(p)\mathbf{C}_{k,p}(p)$, $\mathbf{g}=[\mathbf{g}_0^T,\ldots,\mathbf{g}_{K-1}^T]^T$, where $\mathbf{C}_{k,p-i}(p)$ is a $MN\times(MN+8M-1)$ dimensional matrix as a function of $\beta_{k,p}^n$ that is obtained in details in (9)-(11) of [9]. After whitening, the elements of the noise vector can be deemed as independent and identically distributed (i.i.d.) Gaussian random variables with zero-mean and variance $N_0/2$.

It follows from this discussion that the multiuser detection problem can be cast as 2^K-ary classification problem where the goal is to find the vector of information bits $\mathbf{b}=[b_0(p),\ldots,b_{K-1}(p)]$ given an observation vector $\mathbf{y}(p)$. Assuming all the vectors $\mathbf{b}\in\{0,1\}^K$ are a priori

equiprobable, the minimum distance rule gives the maximum a posteriori decision [10]. Mathematically, the multiuser detection is equivalent to solving the minimization problem $\text{argmin}_{\mathbf{b}\in\{0,1\}^K} \ \mathbf{y}(p) - \sum_{k=0}^{K-1} \mathbf{A}_k(p)\mathbf{g}_k$ in which the term $\mathbf{A}_k(p)$ depends on the choice of $\mathbf{b} \in \{0,1\}^K$. However, the complexity of such detector is exponential in the number of users [10] and in practice sub-optimal methods like minimum mean square error (MMSE) detector [10] are utilized. We consider a case where the BS has access to N_T number of training data from the kth user in the form of $\mathcal{D} = \{\mathbf{y}_i, \mathbf{b}_i\}_{i=1}^{N_T}$, where $\mathbf{y}_i$ has the form of (4) and $\mathbf{b}_i$ denotes the corresponding information bits vector. We further assume that BS does not have access to the perfect knowledge of the true spreading codes from all the users similar to the case of blind MUD [11].

The performance of the MUD algorithms discussed above relies heavily on the estimation of the channel parameters. The joint ML estimate of these parameters is known to require an exhaustive search over the continuous K-dimensional space $[0, T_b)^K$ [9, 12], which is computationally prohibitive. As a workaround, alternative sub-optimal estimation methods of low-complexity are proposed to be used for practical systems. Notably, given the knowledge of the spreading codes and information bits, the authors in [9] propose to directly estimate the overall CIR $\mathbf{g}$ by invoking the LS estimation procedure $\widehat{\mathbf{g}} = \text{argmin}_{\mathbf{x}} \sum_{i=1}^{N_T-1} ||\mathbf{y}_i - \mathbf{A}(i)\mathbf{x}||^2$. Afterwards, relying on $\widehat{\mathbf{g}}$ an ad-hoc algorithm is devised in [9] to estimate the channel parameters, which include delays $\tau_{k,l} = \tau'_{k,l} + \tau_k$, amplitudes $a_{k,l} = A_k|\alpha_{k,l}|$ and phases $\phi_{k,l} = \arg(a_{k,l})$ for $k = 0, \ldots, K-1$ and $l = 0, \ldots, L-1$. In the next section we present numerical results to demonstrate the effectiveness of HyPhyLearn for solving the MUD problem.

5. NUMERICAL RESULTS

We consider a system with processing gain of $N = 32$ where the number of users is set to $K = 3$. Golden codes [9] of length 32 are used by the BS as the pseudo-noise code and the users' amplitudes, A_k's, are set to 2. In addition, a chip interval of length $T_c = 0.001$ and a sampling rate of $2/T_c$ is employed. A near-far ratio (NFR) of 10 dB is assumed, which means the users' amplitudes are randomly unbalanced around 2 with a variance of $\pm$5dB. We further introduce another parameter ρ in order to quantify the averaged error in the pseudo-noise code experienced by the BS while decoding.

As the performance metric, we consider the bit error rate (BER) at the BS while decoding the users' information bits, which is of major interest in digital communication systems. As a multiuser baseline detection algorithm we employ the minimum mean square error (MMSE) decoder introduced in [13], which is shown to outperform other existing detection algorithms. As mentioned in Section 4, MUD can be also solved by a classifier aiming at distinguishing between 2^K different classes, each representing a unique decoded sequence of information bits. In this case, BER is directly related to the classification accuracy of the trained classifier. We present numerical results for the BER performance associated with MUD based on various approaches including HyPhyLearn in Figs. 2 and 3. We note that as the data samples are labeled in the training dataset, the first step of HyPhyLearn is not necessary for this problem. The channel parameter estimation procedure for all the methods is done under two different levels of model mismatch, i.e., $\rho = 0.1$ and $\rho = 0.25$, where the number of available training data from each user is set to 40. Based on these estimates along with the information bits ($\mathbf{b}$) and imperfect knowledge of spreading codes, one can utilize (4) to generate synthetic data.

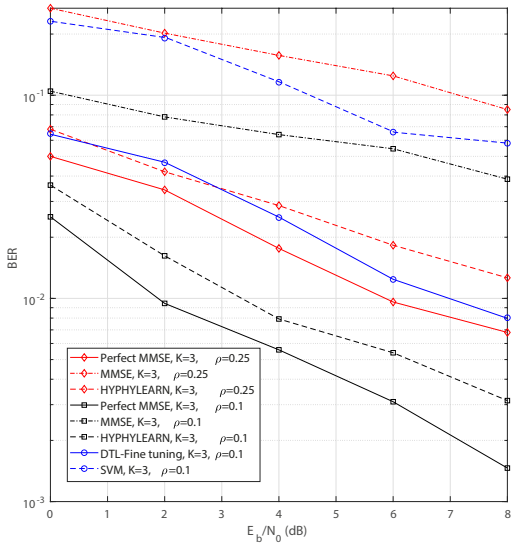

Fig. 2: BER vs SNR for different multiuser detectors.

As a general observation, Fig. 2 demonstrates that the performance of the detectors is deteriorated for a higher value of the PN mismatch parameter ρ. Note that the perfect MMSE is referred to the case where the true pseudo-noise sequences are assumed to be known as part of the implementation of the decoder. In particular, huge performance gap between the perfect MMSE and MMSE decoder indicates the high sensitivity of this detector to the mismatch. Furthermore, the performance of the machine learning algorithm based on support vector machine (SVM) with radial basis function kernel is limited in this case due to limited number of training data. The DTL fine-tuning approach [1] can improve the system performance by training a classifier on the synthetic data and then refining it via the training data. This classifier is chosen to be a DNN with 4 hidden layers of 300 neurons each where ReLU activation function is used for all the hidden layers. Notably, it is highlighted in Fig. 2 that HyPhyLearn outperforms the existing methods over a wide range of SNRs. In Fig. 3, the system performance is investigated as a function of number of available training data, which highlights the superiority of

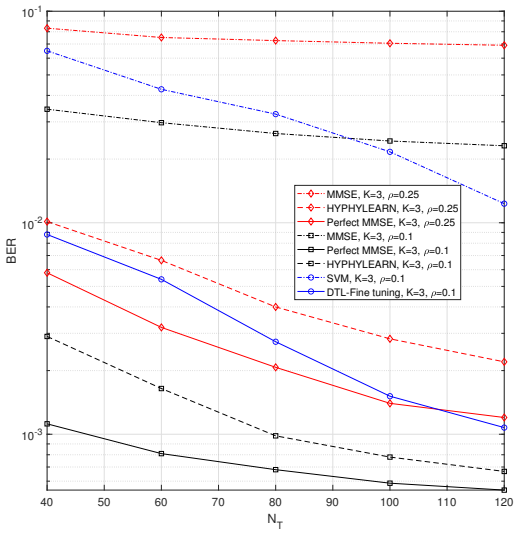

Fig. 3: BER vs the number of available training data at each user.

HYPHYLEARN in the data-limited regime. For this example, SNR at the BS is assumed to be equal to 8 dB. For all the simulations, the number of generated synthetic data is set to 10^6 for both the HYPHYLEARN and fine tuning approaches. Similar to DTL fine-tuning approach, we have used NNs with 4 hidden layers of 300 neurons each for $\mathbf{M}_{\psi}$ and h_{ϕ_1} here. Specifically, h_{ϕ_1} has 2^K output neurons, each corresponding to a specific information bits vector. Also, a shallow NN with one hidden layer of 40 neuron is used for d_{ζ} and ReLU activation function is used for all the hidden layers. During training Adam optimizer with a learning rate of 0.0001 is utilized as the stochastic gradient descent algorithm.

6. CONCLUSIONS

We have revisited the classification problem in a data-limited regime where there is known model for each class while their true parameters are unknown. We have first used (necessarily) suboptimal parameter estimation algorithms for this purpose and generated synthetic data leveraging the knowledge of statistical models. Then, we have utilized the domain adversarial framework for learning a classifier using the synthetic and training data. As a case study, we have considered the problem of multiuser detection, and showed the superiority of our proposed approach in comparison to the several existing statistical and machine learning methods through numerical simulations.

7. REFERENCES

[1] A. Zappone, M. Di Renzo, and M. Debbah, "Wireless networks design in the era of deep learning: Model-based, ai-based, or both?," *IEEE Transactions on Communications*, vol. 67, pp. 7331–7376, 2019.

[2] C. T. Nguyen, N. V. Huynh, N. H. Chu, Y. M. Saputra, D. T. Hoang, D. N. Nguyen, Q. Pham, D. N., E. Dutkiewicz, and W. Hwang, "Transfer learning for future wireless networks: A comprehensive survey," *ArXiv*, vol. 2102.07572, 2021.

[3] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2021.

[4] R. G. Nascimento and F. A. Viana, "Fleet prognosis with physics-informed recurrent neural networks," *ArXiv*, vol. abs/1901.05512, 2019.

[5] M. Chao, Chetan S. Kulkarni, Kai Goebel, and O. Fink, "Fusing physics-based and deep learning models for prognostics," *ArXiv*, vol. abs/2003.00732, 2020.

[6] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, Jan. 2016.

[7] E. L. Lehmann, *Testing Statistical Hypotheses*, Springer Texts in Statistics. Springer, third edition, 2005.

[8] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira, "Analysis of representations for domain adaptation," in *Proceedings of the 19th International Conference on NeurIPS*, Cambridge, MA, USA, 2006, NIPS'06, p. 137–144, MIT Press.

[9] S. Buzzi and V. Massaro, "Parameter estimation and multiuser detection for bandlimited long-code CDMA systems," *IEEE Transactions on Wireless Communications*, vol. 7, no. 6, pp. 2307–2317, 2008.

[10] H.V. Poor and S. Verdu, "Probability of error in MMSE multiuser detection," *IEEE Transactions on Information Theory*, vol. 43, no. 3, pp. 858–871, 1997.

[11] Angela Sara Cacciapuoti, Giacinto Gelli, Luigi Paura, and Francesco Verde, "Widely linear versus linear blind multiuser detection with subspace-based channel estimation: Finite sample-size effects," *IEEE Transactions on Signal Processing*, vol. 57, no. 4, pp. 1426–1443, 2009.

[12] S. Buzzi and H. V. Poor, "On parameter estimation in long-code DS/CDMA systems: Cramer-rao bounds and least-squares algorithms," *IEEE Transactions on Signal Processing*, vol. 51, no. 2, pp. 545–559, 2003.

[13] U. Madhow and M. L. Honig, "MMSE interference suppression for direct-sequence spread-spectrum CDMA," *IEEE Transactions on Communications*, vol. 42, no. 12, pp. 3178–3188, 1994.