Learning with Partial Multi-Outlooks

Jing Chen[1], Yi He[1], Vijay Raghavan
Center for Advanced Computer Studies
University of Louisiana at Lafayette
{jing.chen, yi.he1, raghavan}@louisiana.edu

***Abstract*—How to deal with data that abounds in heterogeneous applications, known as *multi-outlook learning*, is of imperative importance to the creation of general-purpose and flexible intelligent systems. Prior works envision that each data instance appears in all outlooks, however, in practice, it is often the case that every outlook suffers from information incompleteness due to the data availability issue (*e.g.*, privacy concerns). In such a case, existing learning models tend to be fooled by the ambiguous semantics conveyed by the missing outlooks. To fill the gap, we in this paper propose a new learning paradigm, named *Generative Outlook Reproducing via Repository* (GORR), which draws insight from the human analogy of capturing the commonalities among outlooks to reconstruct the missing outlooks. Specifically, GORR leverages the feature relatedness across outlooks to construct an outlook repository. The instances, once being projected onto the outlook repository, would have complete feature representations, where the missing outlooks are generated from the observed ones. Learner trained on the outlook repository then enjoys a complete feature information and thus is capable to perform accurate predictions. Extensive experiments are carried out on both synthetic and real data sets, demonstrating the effectiveness of GORR.**

***Index Terms*—Multi-Outlook Learning, Partial Instance, Missing Data, Heterogeneous Feature Space, Group Lasso**

I. Introduction

Big Data refers to the torrent of information generated by machines and humans that cannot be handled by a typical database. The data is not only big in volume, but also in variety: a range of data types and sources. The data can be represented by multiple, heterogeneous feature spaces, which we refer to as *outlooks*. Consider, for instance, a predictive model for juvenile delinquency rate [1] may assume the totality of students' behaviors. With access to students' demographic data, academic records, mental health, alcohol/drug abuse history, and family environment, we can reveal insights by learning from multiple outlooks, which enable us to answer interesting questions, such as: Are parents' strong opinion of teenagers' alcohol usage helpful to prevent their children from drinking underage while most of their friends have alcohol consumption issues? Will alcohol prevention program for 8th grader help to reduce student risk better than parents' knowledge education and gambling prevention program together based on the limited resource (such as funding)?

Previous research [2]–[7] showed that significant performance improvement can be achieved by multi-outlook learning (MOL) than single outlook learning. The key to success lies in the extraction of a shared feature subspace that *harmonizes* the disparate information stored in multiple outlooks. One flaw of existing approaches is that they all assume the existence of complete information in every outlook. However, this assumption is barely met in practice. For example, students often skip a lot of questions they do not want to face in a survey. Therefore, each outlook suffers from missing pieces of information, resulting in many *partial instances*, *i.e.*, instances with missing outlooks.

A straightforward solution, that enables the adaptation of existing MOL method, is to pad zeros for the partial instances, overlooking information of the missing outlooks. Unfortunately, this solution does not work well in the following sense. Revisit the student data example. Once the zeros show up in the student reports in terms of alcohol usage, it could be the case that either the students do not suffer from alcohol abuse, or they refuse to answer corresponding survey questions. As a result, the zeros convey ambiguous semantics. Involving such ambiguous semantics in the latent subspace extraction could introduce *noises*, which tends to persist and escalate in the later model training phase, leading to the substantial prediction errors. Our empirical study discovers that, with data instances missing in 50% outlooks, this straightforward solution yields an average 54% prediction accuracy in binary tasks – slightly better than coin-tossing.

To address the issue, we in this paper propose a novel learning paradigm, named *Generative Outlook Reproducing via Repository* (GORR), which draws insights from how human experts intelligently handle the partial instances: Human experts are capable to *infer* the ground truth of missing outlooks by using the information of the existing ones. As such, they are unlikely to be fooled by the noisy partial instances. Such a human inference ability mainly results from the experts' experience of leveraging the commonalities among different outlooks. Analogously, GORR constructs an *outlook repository* by capturing the relatedness among features across all outlooks. Given a partial instance, by projecting it onto the outlook repository, its feature representations in the missing outlooks are generated from those in the observed ones. Learner trained on the outlook repository thus enjoys complete information included in both the original and the reconstructed outlooks.

Moreover, to respect the fact that some outlooks encompass useful discriminant information whereas some outlooks could be irrelevant to the label of interest, we embed the Group Lasso (GL) into our learning process. Specifically, during the outlook repository construction process, GL strengthens the

[1]Equal contribution. Y. He and V. Raghavan designed research; J. Chen performed research; Y. He and J. Chen wrote the paper.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

weights of the important outlooks while weakens those of the non-important ones. Thus far, the irrelevant information is abandoned, ensuring that the constructed outlook repository directly serves the learning task of interest. The learning performance hence can be further improved.

Specific contributions in this paper are summarized as follows:

1) We explore a new problem, named *partial multi-outlook learning* (PMOL), which aims to build an accurate model with data instances arbitrarily missing in several outlooks, rather than the prior MOL works that require full feature information from all outlooks.
2) A novel GORR learning paradigm is proposed to solve the PMOL problem. Extensive experiments on 8 benchmark datasets and 2 real-world datasets have been carried out to demonstrate the effectiveness of our proposal.

The rest of this paper is organized as follows. Section II discusses related work. Section III formulates the PMOL problem. Section IV scrutinizes the building blocks of the proposed approach. Section V reports experimental results. We conclude the work in Section VI.

II. Related Work

Our work is closely related to multi-outlook learning, feature reconstruction methods, and feature selection mechanisms. This section discusses the relationships and differences with the existing literature.

Multi-Outlook Learning is also known as multi-view learning, which aims to improve the learning performance in an outlook of interest by leveraging knowledge from multiple auxiliary outlooks. Prior works [2]–[6] envision a consensus pattern matrix that is extracted from the multiple data matrices of disparate outlooks. These methods require full information of all outlooks, which cannot be satisfied in our setting. The method proposed by [8] lifts such a requirement by allowing missing outlooks. However, it imposes two new assumptions: i) there exists at least one outlook that has complete information of all data instances; and ii) a mapping function between the existing and missing outlooks is known in a priori. Unfortunately, both of the new assumptions do not hold in our PMOL problem, and hence [8] cannot be adapted into our scenario. Recent work by [9] further lifts these two assumptions and considers a similar setting as we do. However, it focuses on clustering instead of supervised learning, and therefore it has different technical challenges and solutions.

Feature Reconstruction. As GORR entails learning a reconstructive mapping among outlooks by exploiting feature relatedness, our work is also related to the feature space reconstruction approaches. Specifically, [10]–[12] assume that the most informative subset of features can recover the whole feature space with minimal recovery errors. Therefore, the capability of features to approximate original data is devised as a novel criterion for unsupervised feature selection. Moreover, [13], [14] propose to learn sparse representations of data streams via reconstructing original features from extracted latent features. However, to our best knowledge, none of them explicitly consider the dynamic nature of outlooks and use feature reconstruction method during the learning process.

Feature Selection. To get rid of the redundant information encoded in several outlooks, the feature selection technique is exploited. We refer readers to [15] for a comprehensive literature review on this topic. Roughly, existing feature selection studies can be categorized into wrapper, filter, and embedded methods. In particular, wrapper methods search for a subset of features in a brute-force or strategical fashion and then leverage a predefined learning algorithm to gauge the quality of the selected feature subset. The whole process iterates until a desirable learning performance is obtained. On the other hand, filter methods [16] design various statistical or information-theory based evidence to assess feature importance given a specific label of interest. Nevertheless, both wrapper and filter methods suffer from incomplete information in our partial multi-outlook setting, and hence are not effective. In our approach, we embed the feature selection into the model learning by using Group Lasso [17], [18], inheriting the merits of wrapper and filter methods and meanwhile enjoying a complete feature information via outlook reconstruction.

III. Preliminaries

We begin by summarizing the notational conventions used in this paper. Bold uppercase and lowercase characters are used for matrices (*e.g.*, $\mathbf{A}$) and vectors (*e.g.*, $\mathbf{a}$), respectively. Script typeface is used for sets/spaces (*e.g.*, $\mathcal{A}$). For any matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{A}_{i,j}$ denotes its $(i,j)^{\text{th}}$ entry, and for any vector $\mathbf{a} = [a_1, a_2, \ldots, a_n]^\top \in \mathbb{R}^n$, a_i denotes its i^{th} element. $\|\cdot\|_1$ and $\|\cdot\|_2$ denote the ℓ_1-, ℓ_2-norm, respectively. The Euclidean inner product of two vectors is denoted by $\langle\cdot,\cdot\rangle$.

A. Learning Task Formulation

Suppose we have in total n data instances $\{(\mathbf{x}_j, y_j) \mid j = 1\ldots n\}_{j=1}^n$ and m outlooks $\{\mathcal{X}_i\}_{i=1}^m$, where $\mathcal{X}_i = \mathbb{R}^{d_i}$ denotes the i^{th} outlook. Without loss of generality, we consider that the outlooks are mutually disjoint, *i.e.*, $\mathcal{X}_p \neq \mathcal{X}_q$ for any $p \neq q$, representing different feature spaces (sets). Let $\mathcal{Y}$ be the label space, *i.e.*, $\mathcal{Y} \in \{1, \ldots, C\}$ for classification and $\mathcal{Y} \in \mathbb{R}$ for regression.

Denoted by $(\mathbf{x}_j, y_j) \in \mathcal{X}_1 \times \ldots \times \mathcal{X}_k \times \mathcal{Y}$ the j^{th} training instance that is represented in k outlooks, where $1 \leq k < m$ and k varies in different data instances. The goal of GORR is to find an optimal hypothesis h, such that the empirical risk $\epsilon = \frac{1}{n}\sum_{j=1}^n \ell\big(y_j, h(\mathbf{x}_j)\big)$ is minimized, where ℓ is a predefined loss function such as cross entropy, square loss, *etc*.

B. Generative Outlook Reproducing

Let $\mathcal{R} := \mathcal{X}_1 \cup \ldots \cup \mathcal{X}_m \in \mathbb{R}^{d_1+\ldots+d_k+\ldots+d_m}$ be an *outlook repository* that is a union set of all m outlooks. For each given training instance, a mapping $\phi : \mathbb{R}^{d_1+\ldots+d_k} \mapsto \mathcal{R}$ is learned to generate its feature representations in the unobserved $m-k$ outlooks from the observed k outlooks. Learning ϕ entails the initialization and training of a generative graphical model [19]–[22]. Denoted by $\mathcal{G}$ the graph in which vertices are

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

the features in $\mathcal{R}$. The weight of an edge in $\mathcal{G}$ encodes a relatedness coefficient between a pair of features. For a vertex v, a vector containing the weights of its all out-edges is denoted as θ_v. The graph $\mathcal{G}$ thus can be represented by a matrix $\Theta = [\theta_1, \ldots, \theta_{|\mathcal{R}|}]^\top \in \mathbb{R}^{|\mathcal{R}|\times|\mathcal{R}|}$.

We define $\mathbf{r}_j := [\mathbf{x}_j, \tilde{\mathbf{x}}_j]^\top \in \mathcal{R}$ as the desired representation of the j^{th} training instance in the outlook repository, where $\tilde{\mathbf{x}}_j \in \mathbb{R}^{d_{k+1}+\ldots+d_m}$ denotes its missing feature representation in the unobserved outlooks $\mathcal{X}_{k+1}, \ldots, \mathcal{X}_m$. Training the graphical model is to maximize the following log-likelihood:

$$\mathcal{Q} = \sum_{j=1}^{n} \log \mathbb{P}(\mathbf{r}_j | \mathbf{x}_j, \Theta), \tag{1}$$

where the features in $\mathcal{R}$ are approximated independently as

$$\mathbb{P}(\mathbf{r}_j | \mathbf{x}_j, \Theta) = \prod_{i=1}^{k} \mathbb{P}(\mathbf{r}_j | \mathbf{x}_j^{(i)}, \theta_1, \ldots, \theta_{d_i}), \tag{2}$$

with $\mathbf{x}_j^{(i)}$ being $\mathbf{x}_j$'s representation in the i^{th} outlook.

Following the convention in generative modelling, we model the data generating process in $\mathcal{R}$ with a mixture of Gaussians. Specifically, we let each observed outlook contribute a Gaussian, which makes an intuitive sense since (i) the missing outlooks contribute zero information for approximating $\mathbf{r}_j$; and (ii) the level of information of different outlooks are not equal (*i.e.*, some outlooks are more informative than others). For simplicity, we define $\Phi^{(i)} := [\theta_1, \ldots, \theta_{d_i}]^\top \in \mathbb{R}^{d_i \times |\mathcal{U}|}$. The distribution density function in Eq. (2) is then as:

$$\mathbb{P}(\mathbf{r}_j | \mathbf{x}_j^{(i)}, \Phi^{(i)}) \propto \exp\Big(-\frac{\delta_i}{2}(\mathbf{r}_j - \mathbb{E}(\mathbf{r}_j))^\top \Sigma^{-1} (\mathbf{r}_j - \mathbb{E}(\mathbf{r}_j))\Big), \tag{3}$$

where $\mathbb{E}(\mathbf{r}_j)$ is approximated based on ϕ given $\Phi^{(i)}$ and $\mathbf{x}_j^{(i)}$. Let Σ be a semi-positive definite covariance matrix. Denoted by δ_i the impact of i^{th} Gaussian contributed by the i^{th} outlook, $\sum_{i=1}^{m} \delta_i = 1$.

IV. The Proposed Approach

In this section, we present the building blocks of our proposed approach for solving PMOL problem and discuss the basic ideas behind its design. Section IV-A scrutinizes why and how shall we leverage the label information in the outlook repository construction process. Section IV-B elaborates on our solution to avoid the high-dimensionality of the constructed outlook repository. We unify our solution into an optimization function and propose an algorithm to solve it in Section IV-C.

A. Constructing Outlook Repository Under Supervision

Outlook repository construction (ORC) becomes challenging in our PMOL problem. Directly adapting the outlook reproducing method introduced in Section III-B suffers from information-insufficiency. Specifically, there can exist a sizeable amount of instances that are only observed in very few (in an extreme case, only one) outlook(s) in practice. As a result, the missing outlooks of those instances are unlikely to be correctly recovered given such limited information. During the learning procedure, there could appear arbitrarily many possible $\mathbf{r}_j$'s that perfectly match $\mathbf{x}_j$ on the observable outlooks. Searching the optimal $\mathbf{r}_j$ among all these options requires external information.

In this work, we leverage the plentiful supervision information provided by the class labels to guide the construction of the outlook repository. We aim to obtain $\mathbf{r}_j$ that can help the learner make fewer prediction errors than the original $\mathbf{x}_j$ – in a nutshell, the recovered outlooks, if helpful, should offer extra discriminant power in prediction.

To this end, we design the objective function for the outlook repository construction as below. First, the log-likelihood maximization function is reformulated by joining Eq. (1) and Eq. (2) as:

$$\max \mathcal{Q} = \sum_{j=1}^{n} \sum_{i=1}^{k} \log \mathbb{P}(\mathbf{r}_j | \mathbf{x}_j^{(i)}, \Phi^{(i)}). \tag{4}$$

Without loss of generality, we follow the spirit of [20], [21] to consider a linear approximator, namely $\mathbb{E}(\mathbf{r}_j) = \sum_{s=1}^{d_i} x_s \Phi^{(i)}$ and $\Phi^{(i)} \succcurlyeq 0$ for all i. Then, we insert Eq. (3) into Eq. (4) and obtain:

$$\max \mathcal{Q} = \sum_{j=1}^{n} \sum_{i=1}^{k} -\delta_i \|\mathbf{r}_j - \sum_{s=1}^{d_i} x_s \Phi^{(i)}\|^2. \tag{5}$$

To simplify the further derivation, we define an indicator matrix $\mathrm{diag}(\delta)$ that represents the impact of i^{th} Gaussian *w.r.t.* each feature in each outlook. The $(i,s)^{\text{th}}$ entry of $\mathrm{diag}(\delta)$ is defined as follows:

$$\mathrm{diag}(\delta)_{i,s} = \begin{cases} \delta_i, & \text{if } s^{\text{th}} \text{ feature of } \mathbf{x}_j^{(i)} \text{ exists} \\ 0, & \text{otherwise,} \end{cases}$$

satisfying that $\mathrm{diag}(\delta)\Theta^\top = [\delta_1 \Phi^{(1)}, \ldots, \delta_k \Phi^{(k)}]^\top \in \mathbb{R}^{|\mathcal{R}|\times(d_1+\ldots+d_k)}$. Hence, since $\sum_{i=1}^{m} \delta_i = 1$, the tightest relaxation of Eq. (5) is:

$$\begin{aligned} &\min \sum_{j=1}^{n} \|\mathbf{r}_j - \sum_{i=1}^{k} \delta_i \Phi^{(i)\top} \mathbf{x}_j^{(i)}\|^2 \\ =&\min \sum_{j=1}^{n} \|\mathbf{r}_j - \mathrm{diag}(\delta)\Theta^\top \mathbf{x}_j\|^2. \end{aligned} \tag{6}$$

Surprisingly, by observing Eq. (6), our desired mapping ϕ can be approximated as: $\phi(\mathbf{x}_j) = \mathrm{diag}(\delta)\Theta^\top \mathbf{x}_j$. Now, we define an orthogonal projection operator $\Pi(\cdot)$ such that $\Pi(\mathbf{r}_j) := \mathbf{x}_j$. In other words, $\Pi(\cdot)$ selects the observable outlooks from $\mathcal{R}$. Moreover, we define a linear learning model $\mathbf{w} \in \mathbb{R}^{|\mathcal{R}|}$ that makes predictions based on $\phi(\mathbf{x}_j)$. The final objective function of the outlook repository construction is defined by the following min-max game:

$$\min_{\phi} \sum_{j=1}^{n} \mathcal{L}_{\text{orc}} = \|\mathbf{x}_j - \Pi(\phi(\mathbf{x}_j))\|^2, \tag{7}$$

$$\min_{\mathbf{w}} \max_{\phi} \sum_{j=1}^{n} \mathcal{L}_{\text{sup}} = \ell\big(y_j, \Pi(\mathbf{w})^\top \mathbf{x}_j\big) - \ell\big(y_j, \mathbf{w}^\top \phi(\mathbf{x}_j)\big). \tag{8}$$

Physical Meaning. To facilitate the understanding of the remainder of this work, we here briefly explain the meaning and the intuition behind the design of Eq. (7) and Eq. (8). First,

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

Eq. (7) encourages the consistency of the feature values of the outlooks, before and after the outlook repository construction. Specifically, if the observed outlooks remain unchanged after construction, we accept that the missing outlooks are correctly recovered. A similar idea is adopted in compressed sensing [23] – a matrix with missing values is completed correctly if its visible entries are accurately approximated. Second, Eq. (8) enforces the recovered missing outlooks to be helpful for the learner. The first and second terms in Eq. (8) denote the prediction losses suffered by making predictions on the observed outlooks and on the outlook repository, respectively. On the one hand, we search for the optimal $\mathbf{w}$ that minimizes the prediction losses. On the other hand, we enforce ϕ to maximize the the difference between these two terms – the larger the difference, the more helpful the recovery outlooks in making a prediction. Eq. (7) and Eq. (8) can be jointly optimized by adapting the regularization method, *i.e.*, either of which can be deemed as a regularizer and go into the other's main objective with a negative tradeoff parameter. We shall see this in Section IV-C.

B. Optimization with Outlook Selection

A notable difficulty arises after the outlook repository construction: Even though the learner enjoys complete information provided by the outlook repository, its learning performance can be deteriorated by the irrelevant and redundant features that exist across outlooks, a phenomenon known as *curse of dimensionality*. Revisit the student data example. Since we can handily collect over-abundant information from the students' demographic data, academic records, mental health, alcohol/drug abuse history, family environment, and so forth, storing all data and scanning them multiple times while training leads to storage and computation overheads. It is thus desirable to know which outlooks are more important, such that we can stop collecting data from other outlooks if those important outlooks are capable to offer enough discriminant power. To do this, we propose to adapt Group Lasso [17], [18] to realize the outlook selection idea, with details below.

Denoted by $\mathbf{w}_{\mathcal{X}_i}$ the weight coefficients of the features in the i^{th} outlook, *i.e.*, $\mathbf{w} = [\mathbf{w}_{\mathcal{X}_1}, \ldots, \mathbf{w}_{\mathcal{X}_m}]^\top$. Mathematically, Group Lasso first uses an ℓ_2-norm regularization term for each $\mathbf{w}_{\mathcal{X}_i}$, then it performs an ℓ_1-norm regularization for all previous ℓ_2-norm terms. Intuitively, Group Lasso tends to select or not select features that are from different outlooks as a whole. The penalty function of Group Lasso is formulated as follows:

$$\Omega(\mathbf{w}) = \sum_{i=1}^{m} \delta_i \|\mathbf{w}_{\mathcal{X}_i}\|_2. \tag{9}$$

Note, in Eq. (9), we employ δ_i as a prior to the significance of the i^{th} outlook, which makes a physical sense. A large δ_i means that the corresponding i^{th} outlook plays an important role in recovering other outlooks. If this is the case, Eq. (9) encourages this kind of important outlooks to be selected during the learning process.

Algorithm 1: The GORR algorithm

Input :
PMOL training set $\mathcal{D} = \{(\mathbf{x}_j, y_j)|j = 1, 2, \ldots, n\}$;
The tuning parameters α, λ, and c;
Maximal number of iterations $maxIter$;

Initialize ORC parameters $\phi = \{\delta_i, \Phi^{(i)}\}_{i=1}^m$;
Initialize classifier $\mathbf{w}$;
$iter \leftarrow 1$;
repeat
  $iter \leftarrow iter + 1$;
  for $j = 1, \ldots, n$ **do**
    Initialize a varied step $\eta \leftarrow c\sqrt{1/(iter * n + j)}$;
    if $iter \mod 2 = 0$ **then**
      Optimize $\mathbf{w}$ with fixed ϕ from the last iteration:
      $$\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla_{\mathbf{w}} \mathcal{F}$$
    else
      Optimize ϕ with fixed $\mathbf{w}$ from the last iteration:
      $$\phi \leftarrow \phi + \eta \nabla_{\phi} \mathcal{F}$$
until convergence or $iter$ exceeds $maxIter$;
return classifier $\mathbf{w}$ and trained ORC parameters ϕ

C. Unified Objective Function and Algorithm

By considering (7), (8), and (9) together, our solution to PMOL problem finally reduces to the unified bi-objective function as below:

$$\min_{\mathbf{w}} \max_{\phi} \frac{1}{n} \sum_{j=1}^{n} \mathcal{F} = \mathcal{L}_{\text{sup}} - \alpha \mathcal{L}_{\text{orc}} + \lambda \Omega(\mathbf{w}), \tag{10}$$

with α and λ being two positive tradeoff parameters.

To solve Eq. (10), we adapt the *Blockwise Gradient Descent* (BGD) solver [24], [25]. Following the main steps of BGD, we (i) divide Eq. (10) into two optimization subproblems that are *w.r.t.* $\mathbf{w}$ and ϕ, respectively; and (ii) optimize Eq. (10) by alternating between the two subproblems, optimizing over one while keeping the other one fixed.

We summarize the main steps of our GORR approach in Algorithm 1. The subproblems are optimized in a stochastic fashion, and we set the learning rate η as a varied step, namely, $\eta \propto \sqrt{1/\#iterations}$, which is commonly used in many other gradient-based optimization methods [26]. Intuitively, a small optimization step size leads to a meticulously probing of minimum but requires more iterations for converging. Especially, in our optimization problem, which is non-convex and has many local minima, using a small learning rate may be trapped by the saddle points and incurs inefficient optimization. On the other hand, a large learning rate usually causes overly radical updates, yielding a divergent solution. Thus, we chose a varied learning step, which has been proven by [27] to more effectively lead to convergence.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

V. Experiments

In this section, we begin by introducing the data sets used in this study (Section V-A), followed by the evaluation protocol (Section V-B), and end with presenting the experimental results (Section V-C).

A. Data Sets

We perform the experiments on 10 data sets consisting of 6 synthetic data sets and 4 real-world data sets. Table I summarizes the statistics of the studied data sets.

Synthetic Data are prepared by following the same idea in [20], [28]. We choose 6 UCI data sets [29] that spanned a broad range of domains, including *image*, *text*, *etc*., whose scales vary from 502 to 3782 and dimensions vary from 68 to 1449. These data sets only have one outlook (feature space) at first. We artificially map the original outlook with two random Gaussian matrices, then we have data represented in three outlooks (*i.e*., the original one and two mapped ones). Thereafter, we randomly remove 50% data examples for each outlook while ensuring that each data example appears in at least one outlook.

Linux Kernel Codebase is the first real data set. It comprises 21,193 source code paths from 10 projects written in the C language, including Linux, libc, *etc*. Since each project was built by a separate group, it is reasonable to consider paths from a single project as data instances form an individual outlook. Each outlook encompasses 13 features so that there are 130 features in total. The goal is to predict whether each path is an error path (or not). Please refer to [30] for more details.

Louisiana Juvenile Crime Prevention is the second real data set. This data set was collected through a State-funded project – Communities that Care Youth Survey (CCYS). In the CCYS data set, there are 355 features designed and collected through questionnaires from 8 different prevention areas (*i.e*., 8 outlooks). These features assess students' problematic behaviors and their exposure to a set of scientifically validated risk and protective factors (*e.g*., family, neighborhood, school, peer). In total, 79,988 different teenagers from 6th, 8th, 10th and 12th grades are observed in the data set. Since students are not obligated to finish all the questionnaires, and, in practice, they only answer a few. These students arbitrarily appear or absent in those outlooks. By calculation, only 40% data is effective for further analysis (in synthetic data sets the missing ratio is 50% as fixed). We divided the entire CCYS data sets into three subsets, namely CCYS-A, CCYS-S, and CCYS-V, each of which serves for a particular learning goal. We allow these data subsets to have overlapped data samples. Specifically, we aim to predict the student's academic success, substance abuse, and violence delinquency in CCYS-A, CCYS-S, and CCYS-V, respectively. Such analysis helps us reveal interesting patterns that are critical for the improvement of teenage development. For example, are teenagers living in disorganized, crime-ridden neighborhoods more likely involved in crimes and drug abuse than those living in safe ones? Or, are teenagers being bullied have a higher probability of drop-out? Through experiments, we shall give answers to this kind of questions as shown in Section V-C.

TABLE I
Characteristics of the studied data sets.

Dataset	#examples	#dim. 1*	#dim. 2*	#dim. 3*	Domain
CAL500	502	68	49	70	audio
wdbc	569	30	60	90	image
medical	978	1,449	1,161	1,620	text
Enron	1,702	1,001	974	1,111	text
yeast	2,417	103	85	113	biology
Slashdot	3,782	1,079	981	1,220	text
Linux	21,193	130 (overall in 10 outlooks)			C-Program
CCYS-A	47,293	355 (overall in 8 outlooks)			education
CCYS-S	38,193	355 (overall in 8 outlooks)			education
CCYS-V	24,244	355 (overall in 8 outlooks)			education

* The dimensions of the original, the first mapped, and the second mapped outlooks, respectively.

B. Evaluation Protocol

Baselines. We compare GORR with two MOL learning algorithms, with details follow.

- **MOMAP** [4] is the state-of-the-art MOL algorithm. Its main idea is to learn mappings between pairs of outlooks, such that the model trained on one outlook can be applied to other outlooks so as to improve their prediction performances.
- **HDAMA** [28] solves the MOL problem differently than MOMAP does. HDAMA learns a share latent feature space to summarize the information from all outlooks, meanwhile preserving the manifold sub-structure of each outlook.

Note, neither MOMAP nor HDAMA considers the partial instances. To adapt them to our PMOL problem, we pad zeros for the missing outlooks.

On the other hand, to validate the helpfulness of the missing outlook recovery, we also compare GORR with two feature reconstruction methods.

- **NMF** [31] stands for Nonnegative Matrix Factorization, which has become the *de facto* solution to the matrix completion due to its successes in recommender systems.
- **DLFM** [32] is the state-of-the-art latent factor model that employs a deep-structure for data recovery purposes, rather than NMF which remains to be a shallow model.

For NMF and DLFM, we first concatenate the data matrices of different outlooks into one sparse matrix (*e.g*., the sparseness of the CCYS data matrix is around 60%). Then, we execute the two methods to complete the missing outlooks (matrix entries). To guarantee the fairness of the comparison, we feed the completed matrix into a linear SVM, since GORR also uses a linear classifier.

Metrics. We aim to demonstrate the effectiveness of GORR in both classification and regression tasks. To this end, we

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

employ the classification accuracy and the root mean squared error (RMSE) as the performance metrics.

$$\text{ACC} = 1 - \frac{1}{|\mathcal{T}|}\sum_{t=1}^{|\mathcal{T}|}[\![y_t \neq \hat{y}_t]\!],$$

$$\text{RMSE} = \sqrt{\frac{1}{|\mathcal{T}|}\sum_{t=1}^{|\mathcal{T}|}(y_t - \hat{y}_t)^2},$$

where $\mathcal{T}$ denotes the test data set. The operator $[\![\cdot]\!]$ takes a statement as input and returns 1 if the statement is true. For classification tasks, GORR predicts the label as $y_t = \text{sign}\big(\mathbf{w}^\top\phi(\mathbf{x}_t)\big)$, and for regression $y_t = \mathbf{w}^\top\phi(\mathbf{x}_t)$.

Parameters. For our approach, the optimal parameter sets are grid-searched with respect to the RMSE performance. Specifically, α and λ are both searched in the range of $\{.001, .003, .005, .01, .03, .05, .1, .3, .5\}$. For the compared methods, parameters are set as suggested in the corresponding literature.

C. Experimental Results

This section presents the experimental results, aiming to answer the following research questions.

- Q1 Does our proposed GORR approach outperform the state-of-the-art methods?
- Q2 At what cost can we expect a specific performance gain in making predictions?
- Q3 How effectively can GORR help us analyze real-world problems?

Comparison of Prediction Performance (Q1)

Table II presents the detailed results of performance comparison. GORR shows a significant prediction improvement over the other algorithms on most data sets. Notably, GORR simply adopts a linear classifier as the implementation of the learner. This implies that the performance improvement can be further expected by utilizing a classifier with higher learning capability such as SVM or neural networks. Statistical significance is examined with *paired t-test* at 95% significance level. The win/tie/loss counts of our approach versus the compared methods are summarized in the last row of Table II.

Overall, it is manifested that i) GORR outperforms MOMAP and HDAMA by wining on most data sets; and ii) MOMAP and HDAMA are more sensitive to the dimension change of the data sets than other algorithms. Specifically, MOMAP and HDAMA achieve 63.0% and 70.0% prediction accuracies on three data sets with high dimensions (*i.e.*, medical, Enron, and Slashdot) in average, respectively. The accuracies on the other seven data sets with lower dimensions are 74.7% and 80.2%. Unsurprisingly, the accuracy performances degrade in percentages of 18.6% and 14.6%, respectively. We extrapolate the reasons behind the results as follows. Since MOMAP and HDAMA pad zeros to the missing data, among which the zeros may be noisy and convey ambiguous semantics, the original data distributions are distorted. Such an undesired effect can be escalated in data sets with high dimensions – the higher the dimensionality of a data set, the larger the number of padded zeros. On the contrary, GORR performs more robustly on both high and low-dimensional data sets by achieving 80.3% and 85.7% accuracies, respectively, where only a 6.7% performance degradation is suffered. This finding validates the effectiveness of the outlook repository construction.

NMF and DLFM perform closely on most data sets. In particular, NMF and DLFM achieve 79.1% and 78.9% average accuracies, respectively, both of which outperform MOMAP and HDAMA yet underperform GORR (84.1%). These results are likely to be related to the machinery possessed by NMF and DLFM that can reproduce (recover) the missing outlooks (data). Such a machinery makes them robust to the varying dimension across data sets. This finding further strengthens the necessity and advantage of handling the missing outlooks intelligently rather than simply padding zeros. However, NMF and DLFM split the data recovery and supervised learning into two separate procedures, failing to respect the plentiful label information during the outlook repository construction process. Contrarily, GORR unifies the supervised learning and the outlook repository construction into one optimization objective, and thus shows significant superiority.

Comparison of Runtime Performance (Q2)

A summary of the runtime performance for GORR and the compared methods is reported in Table III. For NMF and DLFM, their runtimes include both the outlook repository constructing and classifier training processes.

Based on the runtime performance comparison results, we make the following observations. First, MOMAP and HDAMA enjoy shorter runtimes on average (228.4 seconds and 219.8 seconds, respectively). Comparing with these two methods, the slowdowns of GORR are within a factor of 1.4 over all data sets. Given that GORR performs significantly better than MOMAP and HDAMA in terms of accuracy, the trade-off on efficiency is acceptable.

Second, the runtimes of GORR (300.4 seconds) are tightly bounded with NMF (324.4 seconds) and much better than DLFM (946.9 seconds). This discrepancy could be attributed to the deep structure of DLFM, in which matrix factorizations are executed in a layer-after-layer fashion. The larger number of learnable parameters makes DLFM difficult to converge in a short period of time. Thereby, though DLFM achieves the closest prediction performance with our GORR, its obvious training time overhead makes it impractical to solve real tasks with tens or hundreds of thousands of data points. On the other hand, NMF enjoys a shallow matrix factorization model and ties our GORR in terms of runtime. Unfortunately, with only one exemption on the wdbc data set, GORR overall outperforms NMF in terms of prediction accuracy. Furthermore, it is worth pointing out that both NMF and DLFM require the data sets to be available beforehand, then the learning is performed in a batch mode. GORR allows data instances to arrive in a stochastic fashion, granting it a higher degree of extensibleness and scalability in real applications. For example, GORR can learn from streaming data while NMF and DLFM need to be re-trained multiple times given incrementally more data. Hence, GORR is more effective and efficient in practice.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

TABLE II

COMPARISON OF GORR WITH BASELINES UNDER TWO EVALUATION METRICES (MEAN ACCURACY ± STANDARD DEVIATION). ↑ AND ↓ INDICATE "THE LARGER THE BETTER" AND "THE SMALLER THE BETTER", RESPECTIVELY. THE BEST PERFORMANCES ARE BOLD. THE BLACK DOT • INDICATES GORR HAS A STATISTICALLY SIGNIFICANT BETTER PERFORMANCE THAN THE COMPARED ALGORITHMS (HYPOTHESIS SUPPORTED BY PAIRED T-TESTS AT 95% SIGNIFICANCE LEVEL).

Dataset	Metric	MOMAP	HDAMA	NMF	DLFM	GORR
CAL500	Accuracy ↑	.773 ± .001 •	.836 ± .000 •	.765 ± .004 •	.802 ± .000	**.881 ± .004**
	RMSE ↓	.667 ± .002 •	.417 ± .000	.562 ± .001 •	.404 ± .001	**.398 ± .002**
wdbc	Accuracy ↑	.834 ± .001 •	.856 ± .001 •	**.939 ± .002**	.893 ± .001	.927 ± .000
	RMSE ↓	.463 ± .000 •	.352 ± .000 •	.180 ± .001	.293 ± .003 •	**.138 ± .000**
medical	Accuracy ↑	.623 ± .002 •	.678 ± .002 •	.739 ± .000 •	.746 ± .003 •	**.806 ± .002**
	RMSE ↓	.814 ± .000 •	.463 ± .000	.628 ± .003 •	.510 ± .001 •	**.407 ± .000**
Enron	Accuracy ↑	.646 ± .000 •	.768 ± .017	.819 ± .002	.787 ± .005	**.845 ± .001**
	RMSE ↓	.686 ± .000 •	.315 ± .000	.421 ± .002 •	.403 ± .012 •	**.308 ± .002**
yeast	Accuracy ↑	.768 ± .000 •	.807 ± .000 •	.812 ± .001	.794 ± .001 •	**.853 ± .001**
	RMSE ↓	.694 ± .000 •	.515 ± .000 •	.419 ± .004	.421 ± .004	**.375 ± .000**
Slashdot	Accuracy ↑	.621 ± .000 •	.653 ± .000 •	.684 ± .010 •	.726 ± .004	**.758 ± .003**
	RMSE ↓	.886 ± .000 •	.542 ± .000 •	.662 ± .009 •	.633 ± .003 •	**.352 ± .077**
Linux	Accuracy ↑	.628 ± .003 •	.745 ± .001 •	.802 ± .006 •	.833 ± .001	**.854 ± .009**
	RMSE ↓	.721 ± .000 •	.423 ± .003	.597 ± .003 •	.442 ± .000	**.403 ± .004**
CCYS-A	Accuracy ↑	.841 ± .001	**.889 ± .000**	.834 ± .003	.796 ± .000	.866 ± .002
	RMSE ↓	.574 ± .000 •	.318 ± .000 •	.377 ± .001 •	.212 ± .002 •	**.303 ± .001**
CCYS-S	Accuracy ↑	.685 ± .000 •	.758 ± .000 •	.742 ± .002 •	.764 ± .001 •	**.811 ± .001**
	RMSE ↓	.706 ± .000 •	.389 ± .000	.506 ± .001 •	.647 ± .003 •	**.350 ± .000**
CCYS-V	Accuracy ↑	.701 ± .001 •	.725 ± .001 •	.778 ± .003	.746 ± .001 •	**.809 ± .000**
	RMSE ↓	.705 ± .000 •	.514 ± .000 •	.689 ± .003 •	.527 ± .001 •	**.436 ± .006**
GORR: w/t/l	Accuracy ↑	9 / 1 / 0	8 / 1 / 1	7 / 2 / 1	6 / 4 / 0	—
	RMSE ↓	10 / 0 / 0	5 / 5 / 0	9 / 1 / 0	7 / 3 / 0	—

TABLE III

COMPARISON OF RUNTIME PERFORMANCE IN THE TRAINING PHASE (IN SECONDS). THE VALUES ARE OBTAINED FROM A 10-FOLD CROSS-VALIDATION.

Dataset	MOMAP	HDAMA	NMF	DLFM	GORR
CAL500	129.75	115.72	282.69	782.69	171.27
wdbc	105.38	111.23	60.74	262.45	160.17
medical	185.99	181.09	155.13	713.91	275.26
Enron	239.52	225.83	242.92	775.28	316.16
yeast	145.11	162.52	188.67	586.02	203.15
Slashdot	240.27	298.13	516.30	1275.26	365.21
Linux	148.12	140.32	136.50	517.43	213.29
CCYS-A	473.40	495.25	837.90	2127.18	643.83
CCYS-S	459.24	320.00	573.67	1374.30	363.96
CCYS-V	156.95	147.62	249.76	1054.80	291.91

Effectiveness of Outlook Selection Mechanism (Q3)

The usefulness of GORR in analyzing real problems presents in twofold. First, GORR recovers the missing outlooks - critical to have a global view of the data set without suffering from missing data. This improvement makes many statistical and data mining tools possible, such as hypothesis tests and decision trees. However, the data incompleteness can also be remediated by several off-the-shelf methods such as Compressed Sensing and Collaborative Filtering. Therefore, we did not pursue further depth in this fold.

Second, and more importantly, as a by-product of GORR's learning process, the impact factors of the outlooks (*i.e.*, δ_i) are good indicators of the importance of a particular outlook over others. This advantage helps us to reveal interesting patterns for a better understanding of real-world problems. We used the CCYS data set as an example to illustrate the effectiveness of the outlook selection mechanism. More details can be found in our technical report [33].

Among 8 outlooks of the CCYS data set, we identify two outlooks that have significant impacts on making predictions, with their corresponding factors are 0.35 and 0.41 (significantly higher than those of other outlooks). The first outlook profiles the teenagers' basic information such as address, demographics, *etc*. This outlook has the least missing data (98.2% data density). The second outlook describes alcohol abuse and prescription drug misuse conditions, which has the highest missing rate (14.1% data density). The result is surprisingly consistent with our previous work [1], [34], [35], showing that (i) Teenagers living in disorganized blocks have higher frequencies in antisocial behaviors with a factor of 1.25; (ii) Adolescent abstainer is 5.6% less for mental health treatment request and 7.5% lower on the depressive level; and (iii) Teenagers who have alcohol abuse and prescription drugs misuse problem are 85% more likely to dropout before the 12-th grade.

While other data completion methods might recover the missing data in the second outlook above, they cannot suggest the importance of this outlook in making predictions. Our

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

GORR approach, contrarily, can simultaneously recover the missing data and indicates more significant outlooks, is thus more useful and helpful in analyzing real-world problems.

VI. Conclusion

In this paper, we explored a novel partial multi-outlook learning (PMOL) problem, where data abound in multiple outlooks, yet each outlook suffers from incomplete instance information. Our key insight is to construct an outlook repository by leveraging the feature relatedness across outlooks. A learner is trained based on the feature representations of the outlook repository, which, on the one hand, enjoys completed feature information, and on the other hand, exploits the plentiful supervised information so as to ensure the helpfulness of the recovered missing outlooks. We showed that our learning problem finally boils down to and can be solved by a unified optimization problem. Experimental results demonstrated the effectiveness and efficiency of our proposal.

In the future, we plan to address the generalizability issue of this work, which might be subject to certain limitations. For instance, the data samples collected from real tasks are usually unlabeled. Human annotation is in general expensive, time-consuming, and error-prone. As a result, further studies need to be carried out in order to investigate how to address the missing outlooks in an unsupervised manner and how to learn an accurate predictive model with scarce labels.

Acknowledgments

The authors would like to thank the IJCNN 2020 reviewers for their constructive feedback. Also, the authors appreciate the Cecil J. Picard Center for being provided with the Louisiana Juvenile Survey data set. This work is supported by the US National Science Foundation (NSF) under grant CNS-1650551.

References

[1] S. J. Dick, C. J. Forsyth, J. Chen, Y. A. Forsyth, R. W. B. Jr., and K. Burstein, "School and peers: Examining the influence of protective factors on delinquency and age of onset," *Deviant Behavior*, vol. 40, no. 4, pp. 476–483, 2019. [Online]. Available: https://doi.org/10.1080/01639625.2018.1438837

[2] C. Hou, C. Zhang, Y. Wu, and F. Nie, "Multiple view semi-supervised dimensionality reduction," *Pattern Recognition*, vol. 43, no. 3, pp. 720–730, 2010.

[3] B. Long, P. S. Yu, and Z. Zhang, "A general model for multiple view unsupervised learning," in *Proceedings of the 2008 SIAM international conference on data mining*. SIAM, 2008, pp. 822–833.

[4] M. Harel and S. Mannor, "Learning from multiple outlooks," in *ICML*, 2010.

[5] L. Duan, D. Xu, and I. Tsang, "Learning with augmented features for heterogeneous domain adaptation," in *ICML*, 2012.

[6] M. Žitnik and B. Zupan, "Data fusion by matrix factorization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 1, pp. 41–53, 2014.

[7] G. Cao, A. Iosifidis, M. Gabbouj, V. Raghavan, and R. Gottumukkala, "Deep multi-view learning to rank," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[8] M. R. Amini, N. Usunier, and C. Goutte, "Learning from multiple partially observed views-an application to multilingual text categorization," in *NeurIPS*, 2009, pp. 28–36.

[9] S.-Y. Li, Y. Jiang, and Z.-H. Zhou, "Partial multi-view clustering," in *AAAI*, 2014.

[10] C. Boutsidis, M. W. Mahoney, and P. Drineas, "Unsupervised feature selection for principal components analysis," in *SIGKDD*. ACM, 2008, pp. 61–69.

[11] J. Li, J. Tang, and H. Liu, "Reconstruction-based unsupervised feature selection: an embedded approach," in *IJCAI*, 2017.

[12] S.-J. Huang, M. Xu, M.-K. Xie, M. Sugiyama, G. Niu, and S. Chen, "Active feature acquisition with supervised matrix completion," *SIGKDD*, 2018.

[13] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *ICML*, 2009, pp. 689–696.

[14] P. Ruvolo and E. Eaton, "Online multi-task learning via sparse dictionary optimization," in *AAAI*, 2014.

[15] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, pp. 1–45, 2017.

[16] D. Wu, Y. He, X. Luo, M. Shang, and X. Wu, "Online feature selection with capricious streaming features: A general framework," in *2019 IEEE International Conference on Big Data*. IEEE, 2019, pp. 683–688.

[17] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[18] F. R. Bach, "Consistency of the group lasso and multiple kernel learning," *Journal of Machine Learning Research*, vol. 9, no. Jun, pp. 1179–1225, 2008.

[19] Y. He, B. Wu, D. Wu, E. Beyazit, S. Chen, and X. Wu, "Online learning from capricious data streams: a generative approach," in *IJCAI*, 2019, pp. 2491–2497.

[20] Y. He, B. Wu, D. Wu, and X. Wu, "On partial multi-task learning," in *ECAI*, 2020.

[21] Y. He, B. Wu, D. Wu, E. Beyazit, S. Chen, and X. Wu, "Toward mining capricious data streams: A generative approach," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[22] M. J. Johnson, D. K. Duvenaud, A. Wiltschko, R. P. Adams, and S. R. Datta, "Composing graphical models with neural networks for structured representations and fast inference," in *NeurIPS*, 2016, pp. 2946–2954.

[23] D. L. Donoho, "Compressed sensing," *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[24] B.-D. Liu, Y.-X. Wang, B. Shen, Y.-J. Zhang, and Y.-J. Wang, "Block-wise coordinate descent schemes for sparse representation," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 5267–5271.

[25] J. Nutini, M. Schmidt, I. Laradji, M. Friedlander, and H. Koepke, "Coordinate descent converges faster with the gauss-southwell rule than random selection," in *ICML*, 2015, pp. 1632–1641.

[26] N. Cesa-Bianchi and G. Lugosi, *Prediction, learning, and games*. Cambridge university press, 2006.

[27] Y. Dauphin, H. De Vries, and Y. Bengio, "Equilibrated adaptive learning rates for non-convex optimization," in *NeurIPS*, 2015, pp. 1504–1512.

[28] C. Wang and S. Mahadevan, "Heterogeneous domain adaptation using manifold alignment," in *IJCAI*, 2011.

[29] D. Dua and E. Karra Taniskidou, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[30] B. Wu, J. P. Campora III, Y. He, A. Schlecht, and S. Chen, "Generating precise error specifications for c: a zero shot learning approach," in *OOPSLA*. ACM, 2019, pp. 160–191.

[31] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *NeurIPS*, 2001, pp. 556–562.

[32] D. Wu, X. Luo, M. Shang, Y. He, G. Wang, and M. Zhou, "A deep latent factor model for high-dimensional and sparse matrices in recommender systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019.

[33] Cecil J. Picard Center for Child Development and Lifelong Learning, University of Louisiana at Lafayette, "2019 Louisiana Caring Communities Youth Survey," Tech. Rep., 2019.

[34] J. Chen, C. J. Forsyth, R. W. Biggar, and K. Burstein, "Determining normal deviance: Adolescent drug use?" *Deviant Behavior*, vol. 40, no. 1, pp. 19–28, 2019. [Online]. Available: https://doi.org/10.1080/01639625.2017.1411007

[35] R. W. Biggar Jr, C. J. Forsyth, J. Chen, and T. A. Richard, "Protective factors for deviance: A comparison of rural and urban youth," *Deviant Behavior*, vol. 37, no. 12, pp. 1380–1391, 2016.