

Modeling Heterogeneous Statistical Patterns in High-dimensional Data by Adversarial Distributions: An Unsupervised Generative Framework

Han Zhang¹, Wenhao Zheng³, Charley Chen¹, Kevin Gao¹, Yao Hu³, Ling Huang² and Wei Xu¹

¹Tsinghua University, Beijing, China

²AHI Fintech, China, ³Youku Cognitive and Intelligent Lab, Alibaba Group

{han-zhan17,ctj2015,kevingao96}@mails.tsinghua.edu.cn,{zwh149850,yaoHu}@alibaba-inc.com,huang.ling@gmail.com,weixu@tsinghua.edu.cn

ABSTRACT

Since the label collecting is prohibitive and time-consuming, unsupervised methods are preferred in applications such as fraud detection. Meanwhile, such applications usually require modeling the intrinsic clusters in high-dimensional data, which usually displays heterogeneous statistical patterns as the patterns of different clusters may appear in different dimensions. Existing methods propose to model the data clusters on selected dimensions, yet globally omitting any dimension may damage the pattern of certain clusters. To address the above issues, we propose a novel unsupervised generative framework called FIRD, which utilizes adversarial distributions to fit and disentangle the heterogeneous statistical patterns. When applying to discrete spaces, FIRD effectively distinguishes the synchronized fraudsters from normal users. Besides, FIRD also provides superior performance on anomaly detection datasets compared with SOTA anomaly detection methods (over 5% average AUC improvement). The significant experiment results on various datasets verify that the proposed method can better model the heterogeneous statistical patterns in high-dimensional data and benefit downstream applications.

CCS CONCEPTS

• **Mathematics of computing** → **Maximum likelihood estimation**; **Probabilistic algorithms**; *Probability and statistics*.

KEYWORDS

unsupervised learning, adversarial distributions, heterogeneous statistical patterns, high-dimensional data, prior knowledge

ACM Reference Format:

Han Zhang¹, Wenhao Zheng³, Charley Chen¹, Kevin Gao¹, Yao Hu³, Ling Huang² and Wei Xu¹. 2020. Modeling Heterogeneous Statistical Patterns in High-dimensional Data by Adversarial Distributions: An Unsupervised Generative Framework. In *Proceedings of The Web Conference 2020 (WWW '20)*, April 20–24, 2020, Taipei, Taiwan. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3366423.3380213>

This paper is published under the Creative Commons Attribution 4.0 International (CC-BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.

WWW '20, April 20–24, 2020, Taipei, Taiwan

© 2020 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC-BY 4.0 License.

ACM ISBN 978-1-4503-7023-3/20/04.

<https://doi.org/10.1145/3366423.3380213>

1 INTRODUCTION

Human-annotated labels are widely adopted to supervise machine learning. Yet, large-scale labeled datasets are usually prohibitive and time-consuming to obtain [8, 20, 28, 34, 36], so the collected labels may be obsolete for training. For example, online platforms label users as fraudsters after detecting certain malicious behaviors. However, once the fraudsters are exposed, they tend to change their strategies of committing fraud so that the platforms cannot utilize these labels to identify fraudsters in the future. As such, unsupervised methods are usually preferred in such applications to perform real-time pattern recognition.

Applications such as fraud detection usually require modeling the intrinsic clusters in high-dimensional data. Such data usually displays *heterogeneous statistical patterns* as the patterns of different clusters may appear in different dimensions. As an example, consider the registration log data in an online platform which contains dozens of dimensions in 4 types: user information (name, ID card number, gender, age, phone number), device information (MAC address, OS type, manufacturer), network information (IP address, channel, browser) and behavior information (timestamp, time elapsed during registration). The fraudsters usually display abnormal *synchronized behaviors* in specific dimensions due to similar control scripts or resource sharing, while the normal users are randomly distributed [24, 27]. For instance, one fraud group may utilize the Android simulator to implement bulk registration, which in turn leads to similar device and behavior information. Another fraud group may hire people to register the accounts manually. As they possess only a few available phone numbers and network proxies, their network and user identity signatures will be similar.

The heterogeneous statistical patterns are challenging to identify, as such fraudsters will try their best to get disguised as normal users. In practice the fraudsters are highly indistinguishable from the normal users concerning all the dimensions due to two reasons: 1) many more dimensions are recorded in real-world scenarios (e.g., more detailed personal information and historical behaviors from other platforms), and 2) normal users may occasionally share feature values. From a global perspective, traditional similarity-based clustering methods may result in too many false positives as they are easily affected by noisy normal users. Many previous works propose to omit uninformative features and cluster the dataset on the remaining features [4, 13, 26]. As the local synchronization of different fraud groups is usually distinct (e.g., device/time information and network/user information), it is often hard to tell which dimension to omit. In fact, the fraud patterns evolve as fraudsters learn

which dimensions are ignored and thus find new ways to avoid detection. Therefore, to effectively recognize the heterogeneous statistical patterns, methods should model both the global distribution of all data clusters and the local patterns for each cluster.

Modeling such heterogeneous statistical patterns is challenging for two reasons. First, globally modeling the data clusters requires evaluating the joint distribution of all dimensions, which suffers from the curse of dimensionality. For example, the number of possible combinations of (gender, age, phone number, ...) is the product of each dimensions' possible value counts. Therefore, the number of parameters required to describe such joint distribution grows exponentially fast with the dimensionality. Second, recognizing the local cluster patterns requires modeling the mixture distribution of the different fraud groups and random normal users, which is usually distinct among the dimensions as the dimensions' sample spaces (i.e., the set of all possible values) are different. For example, the gender and age of a user take values respectively from a binary-value set $\{male, female\}$ and a ten-value set $\{0\sim 10, 10\sim 20, \dots, 90\sim 100\}$, so we need 1 and 9 parameters to describe the distributions on these two dimensions correspondingly. As a result, conventional mixture methods such as Gaussian mixture model (GMM) are inapplicable to such data, since they assume that all dimensions of the data share the same sample space, e.g., $\mathbb{R}$ for GMM.

To address the above issues, we propose a novel generative framework called FIRD to model the heterogeneous statistical patterns in unlabelled datasets. **FIRD** effectively models both the global and local patterns based on the **Feature Independence** assumption and the **adversarial Distributions**. FIRD assumes the features are conditionally independent within the same group to model the dimensions with different sample spaces and relieve the curse of dimensionality. It then utilizes the *adversarial distributions* (a group of distributions competing in generating observations) to fit and disentangle the complex data distributions in each dimension, which in turn brings the model stronger interpretability. Specifically, FIRD identifies the synchronization of the fraud groups as well as the randomly distributed normal users in discrete space by using a pair of adversarial multinomial distributions in each dimension. It is also worth emphasizing that FIRD is not limited to fraud detection tasks. We demonstrate in our experiments that modeling the heterogeneous statistical patterns are also beneficial to anomaly detection tasks (over 5% average AUC improvement over SOTA methods). We expect FIRD to be effective in other applications that model the patterns other than the synchronization and randomness by adopting appropriate adversarial distributions. The learned probabilistic representations also support probabilistic reasoning based on prior information. The major contributions of this paper are distinguished as follows:

- We present a novel generative framework FIRD, which adopts adversarial distributions to capture heterogeneous statistical patterns in unlabeled datasets.
- In discrete spaces, FIRD provides interpretable fraud detection and anomaly detection results on various datasets.
- The effectiveness of FIRD indicates modeling the heterogeneous statistical patterns benefits various downstream tasks.

The remaining of the paper is arranged as follows. We review the related works in Section 2. We introduce the proposed framework

FIRD in Section 3. We demonstrate the experimental results in Section 4. We conclude the paper in Section 5.

2 BACKGROUND

In this section, we review the approaches related to the method and applications discussed in this paper. Then we demonstrate the identifiability issue arisen in the existing generative models.

2.1 Related Approaches

Recognizing the heterogeneous statistical patterns in unlabeled datasets requires modeling both the global distribution of the data clusters and local patterns of each data cluster. When the local patterns of all data clusters are distributed in the same subset of the features, we may omit other non-informative features and perform clustering on the remaining features [4, 13, 26]. The feature selection algorithms in clustering are basically divided into three categories: the filter models, wrapper models, and hybrid models [4]. The filter models use certain criterion to evaluate the quality of the features and then cluster the data w.r.t. high-quality features [39]. The wrapper models enumerate all feature combinations and utilize a specific clustering method to evaluate each combination [16]. Unlike the other two models, the hybrid models simultaneously select useful features and cluster the data points. For example, feature saliency models attempt to fit each feature with either a Gaussian mixture model (GMM) or a global Gaussian [7, 18, 33, 35, 37]. In this way, features fitted by the GMM are effective for clustering, while other features fitted by the global Gaussian are discarded. However, such methods are unable to model the heterogeneous statistical patterns among the dimensions for three reasons. First, since the local patterns of the clusters may involve different dimensions, no dimension should be omitted globally. Second, distributions like multivariate Gaussian assume the dimensions share the same sample space, such as $\mathbb{R}$, which is usually not true in practice. Finally, such model-based methods suffer from an identifiability problem when applying to discrete spaces [33, 37], as they fail to disentangle the clusters with a mixture of multinomials (Section 2.2).

The synchronized fraudsters described above can be seen as dense blocks (i.e., groups of frequently shared feature values) in the dataset. Related works then detect fraud groups by searching for the dense blocks from the dataset filled with randomly distributed normal users. [12] measures a block's density by its likelihood under a Poisson assumption and greedily searches for blocks to optimize it. [30] proposes a greedy optimization framework to optimize a given density measure. D-Cube [31], the successor to [30], has accelerated computation and is thus applicable to much larger datasets. However, these greedy search methods focus only on the fraud patterns, so the changes in the patterns of the normal users can easily affect these models. Specifically, the precision of these methods declines significantly with the growing number of normal users, as the randomly distributed normal users are noise that can significantly interfere with the searching process (Section 4.1.2).

Modeling the heterogeneous statistical patterns can also benefit anomaly detection methods. Anomaly detection methods assume that the majority of a dataset is closely distributed on several specific manifolds [2]. Data points distant from these manifolds are thus

Symbol	Meaning	Symbol	Meaning	Symbol	Meaning
n	index of data	N	total number of data samples	$\mathbf{x}_n$	observed data
g	index of clusters	G	total number of clusters	d_n	hidden cluster indicator
m	index of features	M	total number of features	f_{nm}	adversarial distribution indicator
i	index of feature values	D_m	m -th feature dimension	$\mathcal{F}_m$	m -th dimension
$p(\cdot)$	generative model	$p_k(\cdot)$	k -th generative component	$q(\cdot)$	variational distributions
π_g	mixture weight of g -th cluster	μ_{gm}	adversarial distribution weight	α_{gm}	synchronized multinomial parameter
β_{gm}	random multinomial parameter	ϕ_{ng}	responsibility of g -th cluster	$\tilde{\gamma}_{ngm}$	responsibility of m -th feature
γ_{ngm}	auxiliary variable for E step	$\tilde{\gamma}_{ngm}$	auxiliary variable for E step	λ	hyper parameter for prior regularization

Table 1: Symbols in this paper. The bold uppercase represents a matrix, the bold lowercase represents a vector and the regular type represents a scalar. Parameters with hats (e.g., $\hat{\pi}_g$) are the parameters in the last EM iteration.

identified as anomalies. With this manifold assumption, one can spot the anomalies by linear methods [9, 32], proximity-based methods [10, 11], and outlier ensembles [19, 21, 40]. In high-dimensional datasets, however, only when considering specific dimensions, the anomalies are distant from the manifolds. To select informative features in the outlier detection problem, many previous works propose the ensembles of the base detectors [19, 21, 40]. The basic thinking is resampling the data points and the features so that the model takes merit from different feature combinations and reduces the fraction of anomalies. However, such resampling searches through all feature combinations, which is exponential to the number of dimensions. Besides, when the resampling fails to generate the right feature combination, the detection performance declines significantly.

2.2 Identifiability Problem in Existing Generative Methods

Two model-based feature selection methods are proposed for clustering discrete data [33, 37]. They extend the seminal works [18, 35] from continuous to the discrete case by replacing Gaussian distributions with multinomial distributions. Here we show the identifiability issue of such models in discrete spaces.

To avoid confusion, we use the same denotation as the [33]. The likelihood is

$$\mathcal{L}(\Theta; \mathbf{y}_i) = \prod_{i=1}^N \sum_{k=1}^K \alpha_k \prod_{l=1}^L [\rho_l p(y_{il} | \theta_{lk}) + (1 - \rho_l) q(y_{il} | \theta_l)], \quad (1)$$

where $\mathbf{y}$ is the observation. Parameters α and ρ are the mixture weights and indicator parameters of feature selection, respectively. They use θ_{lk} to represent the group-specific parameters and θ_l to represent the global parameters. Here the distributions p and q are both multinomials. We can rewrite the probability as

$$p(y_{il} = Y_{lm}) = \sum_{k=1}^K \alpha_k [\rho_l \theta_{lkm} + (1 - \rho_l) \theta_{lm}], \quad (2)$$

where θ_{lkm} and θ_{lm} are the probability mass of Y_{lm} . Then we can write the likelihood in a different way as

$$\mathcal{L}(\Theta; \mathbf{y}_i) = \prod_{i=1}^N \prod_{l=1}^L \prod_{m=1}^{M_l} \left\{ \sum_{k=1}^K \alpha_k [\rho_l \theta_{lkm} + (1 - \rho_l) \theta_{lm}] \right\}^{\mathbf{1}_{\{y_{il}=Y_{lm}\}}}, \quad (3)$$

where the function $\mathbf{1}_{\{\cdot\}}$ is the indicator function. If we have already learned the optimal ρ^* , and denoting

$$\tilde{p}_{lm} = \sum_{k=1}^K \alpha_k [\rho_l^* \theta_{lkm} + (1 - \rho_l^*) \theta_{lm}],$$

we see that optimizing such mixture of multinomials is equivalent to optimizing a single multinomial with parameter $\tilde{\mathbf{p}}_l = [\tilde{p}_{l1} \dots \tilde{p}_{lm} \dots \tilde{p}_{lM}]$ for each feature. If we introduce information criteria to determine G automatically as they did in the original method, G will always be 1. The existing methods introduce informative priors merely to the mixture weights and feature selection indicator parameters, which does not solve the identifiability issue.

3 OUR METHOD

In this section, we first present a novel unsupervised generative learning framework called FIRD and describe how it captures the heterogeneous statistical patterns using the adversarial distributions. Then we apply the framework to discrete space to model two specific patterns, namely synchronization, and randomness, for applications such as fraud detection and anomaly detection. For better readability, Table 1 summarize the meanings of the symbols in this paper.

3.1 FIRD: A Novel Learning Framework

In this section, we present a novel unsupervised generative learning framework called FIRD. We first introduce the adversarial distributions and discuss the identifiability issue caused by over-flexible variational distributions. Then we propose to solve this identifiability problem using prior knowledge and provide theoretical guarantees.

For each observation in $\mathcal{D} = \{\mathbf{x}_n\}_{n=1}^N$, we want to learn its associated latent semantic variable d_n . To characterize the different sample space and relieve the curse of dimensionality, we assume the data features are independent given d_n , i.e. $p(\mathbf{x}_n | d_n) = \prod_{m=1}^M p(x_{nm} | d_n)$. We assume the observations display up to K distinct statistical patterns in each dimension. To characterize and balance these patterns, we introduce the *adversarial distributions* $\{p_k(x_{nm} | d_n)\}_{k=1}^K$, such that given d_n , the distribution of observation x_{nm} is a mixture of the adversarial distributions, i.e. $p(x_{nm} | d_n) = \sum_{k=1}^K \mu_{mk} p_k(x_{nm} | d_n)$. Here μ_{mk} determines the responsibility of each adversarial distribution component $p_k(x_{nm} | d_n)$ for generating x_{nm} . As μ_m lies in a

probability simplex, increasing the responsibility μ_{mk} of $p_k(x_{nm}|d_n)$ will reduce that of other adversarial distribution components. In this way, the adversarial distributions compete in generating the observations by fitting the intricate data patterns in each dimension. Under such assumption, the data generation process can be described by the following two steps:

- (1) Generate d_n from the semantic distribution $p(d_n)$.
- (2) For each feature m , choose an adversarial component k w.r.t. μ_{mk} , and generate x_{nm} from $p_k(x_{nm}|d_n)$.

Since $p(d_n)$ and $p_k(x_{nm}|d_n)$ are unknown, we cannot directly evaluate the likelihood of such generative models. We then turn to optimizing the likelihood lower bound using variational methods. Denoting the parameters of the learner as π and θ , we have the evidence lower bound of the log-likelihood (ELBO):

$$\begin{aligned} \log \mathcal{L}(\theta, \mathbf{d}; \mathbf{x}) &= \sum_{n=1}^N \log \sum_{d_n} p(d_n|\pi) \prod_{m=1}^M \sum_{k=1}^K \mu_{mk} p_k(x_{nm}|d_n, \theta) \\ &\geq \sum_{n=1}^N \sum_{d_n} q(d_n) \left\{ \log \frac{p(d_n, \pi)}{q(d_n)} + \sum_{m=1}^M \sum_{k=1}^K \tilde{\mu}_{nmk} \log \frac{\mu_{mk} p_k(x_{nm}|d_n, \theta)}{\tilde{\mu}_{nmk}} \right\}, \end{aligned} \quad (4)$$

where we introduce variational distribution $q(d_n)$ to approximate the posterior $p(d_n|\mathbf{x}_n)$ and apply Jensen's inequality with auxiliary variables $\tilde{\mu}_{nmk}$ ($\sum_k \tilde{\mu}_{nmk} = 1, \tilde{\mu}_{nmk} \geq 0$).

As the ELBO is concave, we can use the EM algorithm to iteratively approximate the latent semantic distribution $p(x_n|d_n)$ and estimate the parameters. However, if the adversarial distributions are too flexible, we may fail to disentangle the complex data distribution as expected. An extreme case is that if $p_k(x_{nm}|d_n, \theta)$ is able to fit the data patterns perfectly, i.e. $\inf_{\theta} D_{KL}(p_k(x_{nm}|d_n, \theta) \| p^*(x_{nm}|d_n)) = 0$, it can dominate the adversarial distributions with $\mu_{mk} = 1$ and $\mu_{mj} = 0$ for all $j \neq k$, where we denote the ground truth distributions as $p^*(\cdot)$. To address this problem, we introduce prior information into these adversarial distributions such that each adversarial component can fit one specific pattern much better than other patterns. We have the following theorem to guarantee the optimality and uniqueness of the variational approximation.

THEOREM 1. Suppose that with proper priors, $\forall k \in \{1, \dots, K\}$ there exists a unique and distinct index $k' \in \{1, \dots, K\}$ and some constant $C > 0$ such that

$$\begin{aligned} \inf_{\theta} D_{KL}(p_{k'}^*(x_{nm}|d_n) \| p_j(x_{nm}|d_n, \theta)) &> C, \\ \inf_{\theta} D_{KL}(p_{k'}^*(x_{nm}|d_n) \| p_k(x_{nm}|d_n, \theta)) &= 0 \end{aligned}$$

for all $j \neq k$, where $D_{KL}(\cdot \| \cdot)$ is the KL divergence. Then we have the unique optimal solution:

$$\begin{aligned} q(d_n) &= p^*(x_{nm}|d_n), \quad p_k(x_{nm}|d_n, \theta) = p_k^*(x_{nm}|d_n) \\ \tilde{\mu}_{nmk} &= \frac{\mu_{mk} p_k^*(x_{nm}|d_n)}{\sum_{k=1}^K \mu_{mk} p_k^*(x_{nm}|d_n)} \end{aligned}$$

for the ELBO in Eq. (4).

PROOF. Please see Appendix A. $\square$

To address the specific modeling problem in applications such as fraud detection, we apply FIRD to the discrete spaces.

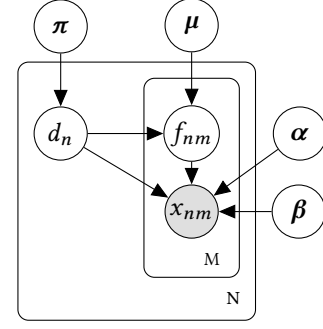


Figure 1: The plate representation of FIRD in discrete space. The plates with subscript M and N indicate respectively the M independent features and N i.i.d. data points. The parameter π is the mixture weight. μ balances the adversarial distribution pairs. The synchronization and randomness are captured by the adversarial distribution pairs, whose parameters are α and β , respectively.

3.2 Discrete Space Application

In this section, we apply FIRD to discrete spaces for applications such as fraud detection. We first illustrate the model and optimization techniques. After that, we describe how FIRD reduces the noise in the dataset by filtering outliers. Then we introduce how FIRD makes inference based on the prior knowledge and the learned probability representations. Finally, we discuss the complexity, initialization, and hyperparameters.

3.2.1 Model. The FIRD model in discrete space is a finite mixture model with adversarial multinomial distribution pairs. To better illustrate the generation process, we plot the plate representation of the FIRD in discrete spaces in Figure 1.

In discrete spaces, the observations $\mathcal{D} = \{\mathbf{x}_n\}_{n=1}^N$ are M -dimensional vectors, where for each feature $\mathcal{F}_m$, $1 \leq m \leq M$, x_{nm} takes its value from the set $\{X_{mi}\}_{i=1}^{D_m}$. The latent semantic variable d_n takes discrete values from $\{1, \dots, G\}$ governed by $p(d_n) = \text{Mult}(\pi)$, where we use $\text{Mult}(\cdot)$ for abbreviation of a multinomial distribution. We assume that given d_n , each feature $\mathcal{F}_m$ independently generates the feature value from an adversarial multinomial distribution pair that respectively captures synchronization and randomness. Suppose the synchronization-aware components of the adversarial distribution pairs are multinomials controlled by α , and the randomness-aware components are multinomials controlled by β . We then have the following generation process for an observation $\mathbf{x}_n$:

- (1) Choose the semantic variable $d_n \sim \text{Mult}(\pi)$.
- (2) For each feature $\mathcal{F}_m$:
 - (a) Choose the indicator variable $f_{nm} \sim \text{Bernoulli}(\mu_{d_n m})$.
 - (b) If $f_{nm} = 1$, choose the observed value $x_{nm} \sim \text{Mult}(\alpha_{d_n m})$;
 - (c) If $f_{nm} = 0$, choose the observed value $x_{nm} \sim \text{Mult}(\beta_{d_n m})$.

As described in section 3.1, we need to introduce priors into the adversarial distributions so that each component of the adversarial distributions fits one specific pattern. In discrete spaces, to generate synchronized observations, the probability mass of the distribution $\text{Mult}(\alpha)$ should distribute on a small subset of the feature values. In other words, most entries in α should be zero so that sampling from

$Mult(\alpha)$ leads to several possible outputs. Similarly, to model the randomness $Mult(\beta)$ should approximate the uniform distribution, so that sampling from $Mult(\beta)$ produces a wide variety of values. We can achieve such randomness- and synchronization-awareness by introducing Dirichlet-like priors to promote sparse $\alpha_{d_n m}$ and smooth $\beta_{d_n m}$. Moreover, according to the traditional information criterions, we can promote sparse π to automatically choose a proper G for the dimension of d_n (see for example [5]).

$$\begin{aligned} & \log \mathcal{L}(\pi, \mu, \alpha, \beta; \mathcal{D}, \lambda^{(1)}, \lambda^{(2)}) \\ &= \sum_{n=1}^N \log \left\{ \sum_{d_n, f_n} p(\mathbf{x}_n | f_n, d_n, \alpha, \beta) p(f_n | d_n, \mu) p(d_n | \pi) \right\} \\ & - \sum_{g=1}^G \lambda_g^{(1)} \log \pi_g - \sum_{g=1}^G \sum_{m=1}^M \sum_{i=1}^{D_m} \lambda_{gmi}^{(2)} (\log \alpha_{gmi} - \log \beta_{gmi}). \end{aligned} \quad (5)$$

We introduce the hyper parameter λ to control the regularization. Note that the regularizers for π and α (last two terms in Eq. (5)) are negative to promote sparsity, and that for β is positive to promote randomness. We choose the same weight $\lambda_{gmi}^{(2)}$ for each adversarial distribution pair to ensure they are equally capable of modeling the synchronization and randomness. We can determine whether $Mult(\alpha)$ or $Mult(\beta)$ prevails by introducing priors to parameter μ , which makes FIRD fully Bayesian. However, in this paper, we just adopt a fair competition between the adversarial distributions.

3.2.2 Optimization. We then apply the EM algorithm for optimization. In the E-step, we calculate the posterior distribution of the latent variables given the observations, i.e. $p(d_n, f_n | \mathbf{x}_n, \hat{\theta})$. To represent this posterior distribution we estimate

$$\begin{aligned} \tilde{\phi}_{ng} &= p(d_n = g | \mathbf{x}) = \frac{\hat{\pi}_g \prod_{m=1}^M \{\gamma_{ngm} + \tilde{\gamma}_{ngm}\}}{\sum_{g'=1}^G \hat{\pi}_{g'} \prod_{m=1}^M \{\gamma_{ng'm} + \tilde{\gamma}_{ng'm}\}}, \\ \tilde{\gamma}_{ngm} &= p(f_{nm} = 1 | \mathbf{x}, d_n = g) = \frac{\gamma_{ngm}}{\gamma_{ngm} + \tilde{\gamma}_{ngm}}, \end{aligned} \quad (6)$$

where we denote the parameter estimations from the last iteration with hats and define

$$\gamma_{ngm} = \hat{\mu}_{gm} \prod_{i=1}^{D_m} \hat{\alpha}_{gmi}^{x_{nmi}}, \tilde{\gamma}_{ngm} = (1 - \hat{\mu}_{gm}) \prod_{i=1}^{D_m} \hat{\beta}_{gmi}^{x_{nmi}}.$$

In the M-step, we find parameters that optimize the expected likelihood. Since the objective is concave w.r.t. μ and β , setting the derivative to zero gives us

$$\begin{aligned} \mu_{gm} &= \frac{\sum_{n=1}^N \tilde{\gamma}_{ngm} \cdot \tilde{\phi}_{ng}}{\sum_{n=1}^N \tilde{\phi}_{ng}}, \\ \beta_{gmi} &= \frac{\lambda_{gmi}^{(2)} + \sum_{n=1}^N x_{nmi} (1 - \tilde{\gamma}_{ngm}) \tilde{\phi}_{ng}}{D_m \lambda_{gmi}^{(2)} + \sum_{n=1}^N (1 - \tilde{\gamma}_{ngm}) \tilde{\phi}_{ng}}. \end{aligned} \quad (7)$$

The optimization of π and α does not have a closed-form solution, as the objective is no longer concave due to the regularizers. However, by introducing some small noise to the Dirichlet-like priors, we can calculate the maximum using a numerical method [17]. We

iteratively update π and α

$$\begin{aligned} \pi_g &= \frac{\sum_{n=1}^N \tilde{\phi}_{ng} + \lambda_g^{(1)} \pi_g}{N + \lambda_g^{(1)} / \pi_g}, \\ \alpha_{gmi} &= \frac{\sum_{n=1}^N x_{nmi} \cdot \tilde{\gamma}_{ngm} \cdot \tilde{\phi}_{ng} + \lambda_{gmi}^{(2)} \alpha_{gmi}}{\sum_{n=1}^N \tilde{\gamma}_{ngm} \cdot \tilde{\phi}_{ng} + \lambda_{gmi}^{(2)} / \alpha_{gmi}}, \end{aligned} \quad (8)$$

until convergence. We present the detailed derivation in the supplementary material¹, and we show the pseudo-code of FIRD in Algorithm 1.

Algorithm 1 FIRD in Discrete Space

Input: Observations $\mathcal{D} = \{\mathbf{x}_n\}_{n=1}^N$, semantic dimension G , regularization weight λ , precision ϵ .
Output: Semantic representations $\tilde{\phi}$ and $\tilde{\gamma}$, semantic distribution π , balance parameters μ of the adversarial distributions, adversarial distribution parameters α and β .

- 1: **for** $g = 1$ to G , $m = 1$ to M **do** ▷ Initialize parameters.
- 2: $\pi_g \leftarrow 1/G, \mu_{gm} \leftarrow 0.5$.
- 3: Randomly initialize and normalize α_{gm} and β_{gm} .
- 4: **end for**
- 5: $\mathcal{L} \leftarrow -\infty$ ▷ Initialize likelihood.
- 6: **repeat**
- 7: Evaluate $\mathcal{L}^{new}$ according to Eq. (5).
- 8: Calculate $\tilde{\phi}, \tilde{\gamma}$ according to Eq. (6). ▷ E-step.
- 9: Update μ and β according to Eq. (7). ▷ M-step.
- 10: Update π and α according to Eq. (8). ▷ M-step.
- 11: **until** $\mathcal{L}^{new} - \mathcal{L} < \epsilon$

3.2.3 Noise Reduction. Since the real-world datasets always contain a considerable amount of noisy data points, here we describe how FIRD deals with such noisy datasets by filtering the outliers.

Observations that do not belong to any possible values of the latent semantic variable are recognized as outliers, which are noise to be removed from the dataset. Specifically, the likelihood of the observation given $d_n = g$ is

$$p(\mathbf{x}_n | d_n = g) = \prod_{m=1}^M \{\gamma_{ngm} + \tilde{\gamma}_{ngm}\}.$$

Given $d_n = g$, the information gain after observing $\mathbf{x}_n$ is

$$I(\mathbf{x}_n | d_n = g) = -\log p(\mathbf{x}_n | d_n = g).$$

We expect this information to be large for outliers, so we compute a threshold on I . A natural choice is its expectation, i.e. the entropy of the distribution $p(\mathbf{x}_n | d_n = g)$:

$$\begin{aligned} H[p(\mathbf{x}_n | d_n = g)] &= \sum_{m=1}^M H[p(x_{nm} | d_n = g)] \\ &= - \sum_{m=1}^M \sum_{i=1}^{D_m} h(\mu_{gm} \alpha_{gmi} + (1 - \mu_{gm}) \beta_{gmi}), \end{aligned} \quad (9)$$

¹Supplementary material is available at https://github.com/fingertap/fird.cython/blob/master/suppl_material.pdf.

where we defined $h(y) = y \log y$. Then we can filter out the outliers that satisfy $I(\mathbf{x}_n | d_n = g) > (1 + \epsilon) \cdot H[p(\mathbf{x}_n | d_n = g)]$ for all components, where ϵ is the tolerance.

3.2.4 Inference. We can incorporate our prior knowledge of $\mathbf{d}$ to make inference based on the learned probabilistic representations. Using classification as an example, we can infer the label of each data point given the labels of $\mathbf{d}$:

$$\ell_n \triangleq \mathbb{E}_{d_n} [\ell | \mathbf{x}_n] = \sum_{g=1}^G p(\ell | d_n = g) p(d_n = g | \mathbf{x}_n), \quad (10)$$

where ℓ is application-dependent target variable. Since $G \ll N$, domain experts can efficiently analyze the learned latent representations $\mathbf{d}$ and design appropriate decision distributions $p(\ell | d_n = g)$.

In fraud detection, we have the prior knowledge that fraudsters usually synchronize with each other due to potential resource sharing [24, 27]. This allows us to determine $p(\ell | d_n = g)$ automatically by calculating the “difficulty of generation” under a random model $\mathcal{M}_{\text{random}}$. If the observations are too difficult to generate under the random model, they are very likely to be fraudsters. Suppose the values X_{mi} have probability $\frac{1}{D_m}$ for each $\mathcal{F}_m$ in the random model $\mathcal{M}_{\text{random}}$. Then the probability of generating g is

$$p(d_n = g | \mathcal{M}_{\text{random}}) = \prod_{n=1}^N \pi_g \prod_{m=1}^M \prod_{i=1}^{D_m} \binom{D_m}{N_{gmi}} D_m^{-N_{gmi}},$$

where we have defined

$$N_{gmi} = \sum_{n=1}^N \mathbf{1}(x_{nm} = X_{mi}) \cdot p(d_n = g | \mathbf{x}_n).$$

Note that $\binom{D_m}{N_{gmi}}$ can be calculated by the gamma function, and $\mathbf{1}(\cdot)$ is the indicator function. Again, we can calculate the information and use the entropy as a threshold for fraudsters:

$$H[d_n = g | \mathcal{M}_{\text{random}}] = \sum_{n=1}^N \sum_{m=1}^M \sum_{i=1}^{D_m} \left\{ \log \binom{D_m}{N_{gmi}} - N_{gmi} \log D_m \right\}.$$

Any group with $I(d_n = g | \mathcal{M}_{\text{random}}) > (1 + \epsilon) \cdot H[p(d_n = g | \mathcal{M}_{\text{random}})]$ is recognized as fraud, where ϵ is the tolerance.

3.2.5 Discussion. We first discuss the complexity of FIRD. FIRD runs in linear time w.r.t. the feature dimension M and the number of data points N . In the E-step, we need to calculate the information of the latent variables, which is $O(NGM)$ for time and space complexity. In the M-step, the time complexity bottleneck is updating α and β . A plain implementation of the update iterations for α and β runs in $O(NG \sum_m D_m)$. However, suppose $NM > \sum_m D_m$, we can trade space for time by maintaining an array for α_{gm} and β_{gm} . With additional $O(\max_m D_m)$ space, the time cost for updating α and β can be reduced to $O(NGM)$. Therefore, both the time and the space complexities of FIRD are $O(NGM)$. When the number of samples is extremely large, a single pass of E step and M step is time-consuming. We can further utilize stochastic EM algorithms to accelerate convergence for large datasets [6, 23].

We also provide a parameter initialization strategy. FIRD adopts pairs of adversarial distributions. If one distribution prevails due to the random initialization (for example, $\mu_{gm} > 0.5$ means the sparse distribution prevails, and $\mu_{gm} = 1$ is equivalent to removing

$\text{Multinomial}(\beta_{gm})$), this advantage may persist through the subsequent iterations. Therefore, a fair start is $\pi_g = 1/G$ and $\mu_{gm} = 0.5$ for all possible g, m . The randomness of the model then comes from the random initialization of α and β .

The hyperparameter λ is hard to determine without any upper bound. Here we discuss how to normalize its value to ease the choice of hyperparameters. λ is the conjugate priors’ weight parameter, which controls the sparsity in α and π as well as the randomness in β . We expect such regularizers to be large enough to punish the models that violate our assumptions but are not so large that they overpower the likelihood. Since the priors have the same form as the likelihood, the regularizers can be seen as the likelihood of fake data points. With a given dataset, we then adjust λ so that the fake data points have a comparable size, i.e.

$$\lambda_g^{(1)} = \lambda^{(1)} \cdot \frac{N}{G}, \quad \lambda_{gmi}^{(2)} = \lambda^{(2)} \cdot \frac{N}{2GD_m}. \quad (11)$$

Thus, we just need to decide the normalized regularization weights $0 < \lambda^{(1)}, \lambda^{(2)} \leq 1$.

4 EXPERIMENTS

In this section, we present detailed analysis and two applications of FIRD. We first demonstrate the application of FIRD on fraud detection. We compare FIRD to the state-of-the-art unsupervised fraud detection methods on an E-commerce platform dataset and visualize the detection results of FIRD as critical applications like fraud detection require high interpretability. Then we report the performance of FIRD on anomaly detection benchmark datasets to show its effectiveness as a general anomaly detection method. Finally, we analyze how the hyperparameters of FIRD affect the performance and its running time cost on synthetic datasets.

We do not compare FIRD with supervised methods for fraud detection as they offer little practical value. Three primary reasons are: 1) the fraud labels are expensive to collect; 2) the fraud pattern changes as the detection method evolves; and 3) theoretically, the distributions of training data and test data may differ vastly for the two reasons above, which violates the i.i.d. assumption of supervised methods.

Since FIRD models both the synchronization and randomness patterns, it displays superior performance on different tasks. As such, in our experiments, we compare with the most promising methods on each task. Specifically, for the fraud detection experiment, we compare with the dense block detection methods M-Zoom, M-Biz, and D-Cube [30, 31], which heuristically search for high-density data blocks (with specially designed density definitions). For anomaly detection methods, we compare FIRD to the state-of-the-art methods such as the histogram-based outlier score (HBOS) [10], the isolation forests (IForest) [21], the one-class SVM (OCSVM) [9] and the clustering-based local outlier factor (CBLOF) [11]. For all these experiments, we adopt a Cython implementation of FIRD².

4.1 Identify Fraudsters in E-commerce Platform

In this experiment we first describe the experiment setups. Then we present the comparison result of FIRD with state-of-the-art fraud

²A Cython implementation is available at <https://github.com/fingertap/fird.cython>.

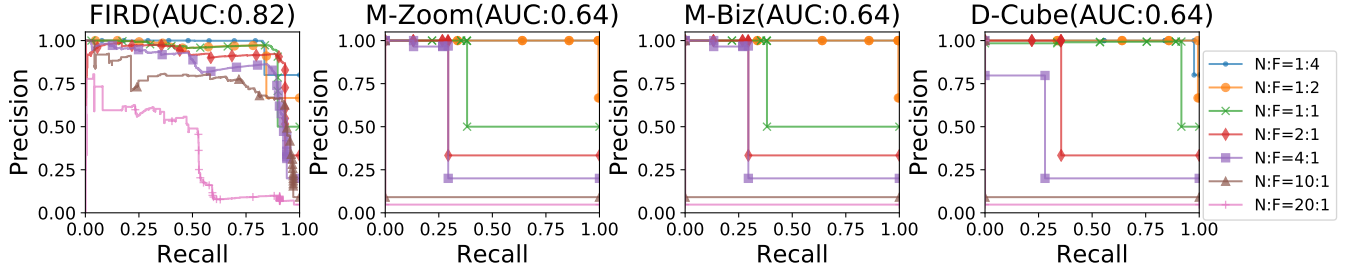


Figure 2: The precision-recall curves on the real-world E-commerce platform data. We present the comparison results with different NFR and plot precision-recall curves. For each method, we evaluate the overall performance by AUC.

detection methods on an E-commerce platform dataset. Finally, we visualize the probabilistic representations learned by FIRD since models are expected to make interpretable decisions in critical applications like fraud detection.

4.1.1 Experiment Setup. Every day, a massive number of new users register for E-commerce platforms, a considerable number of which are fraudsters. We obtained a dataset with over 20,000 fraudsters and a sufficient number of normal user samples with 30 useful features from an E-commerce platform. For some features, there are tens of thousands of possible values. The platform labeled the records according to the account behavior in the following few months. As previously described, the fraudsters form different clusters by sharing different sets of feature values. In contrast, normal users seldom share feature values and are randomly distributed.

We randomly sample 7 sets of normal users with different sizes and mix them with the fraudsters to synthesize seven datasets with different normal-user-to-fraudster ratios (NFR, ranging from 1:4 to 20:1). These datasets have the same fraud patterns, and the main difference lies in the noise levels. We apply FIRD on these datasets and compare its performance to the existing dense block detection methods M-Zoom, M-Biz [30], and D-Cube [31]. We use these methods to predict the identities of the data points and evaluate the precision and recall. We decide $p(\ell|d_n = g)$ for FIRD according to the method described in section 3.2.4. For the dense block detection methods, we assign $p(\ell|d_n = g) = 1$ if the proportion of the fraudsters in the detected block exceeds 50%, and assign the labels of the detected blocks to their members. FIRD labels the outliers filtered out according to Section 3.2.3 as normal.

4.1.2 Overall Comparison Results. We display the comparison results in Figure 2. Note that for the dense block methods, we report the best result under four different density measures. We plot the precision-recall curve and present the mean area under curve (mean AUC) score for all four methods on these seven datasets. We observe that, for dense block detection methods, there is a precision decline as the NFR increases. The AUC score of these methods decreases rapidly as the NFR increases (from 0.99 to 0.1). For comparison, we observe that FIRD is relatively robust to the NFR (AUC ranges from 0.97 to 0.46, with an average of 0.82). These improvements can be attributed to the refining process, which deletes most of the noisy data points.

As critical applications like fraud detection expect the model to provide interpretable results, we visualize FIRD's detection results with the following concrete example.

4.1.3 Interpretable Results. In fraud detection applications, the model is expected to provide interpretable results. As such, we visualize the probabilistic representations learned by FIRD to show how it captures the fraud patterns. We first plot the distribution of fraudsters and normal users after detection. Then we investigate one specific case, i.e., $d_n = 1$, to show FIRD captures the fraud pattern. Finally, we demonstrate the IP address distribution given $d_n = 1$ to show that FIRD captures the synchronizations in both fraudsters and some (potentially malicious) normal users.

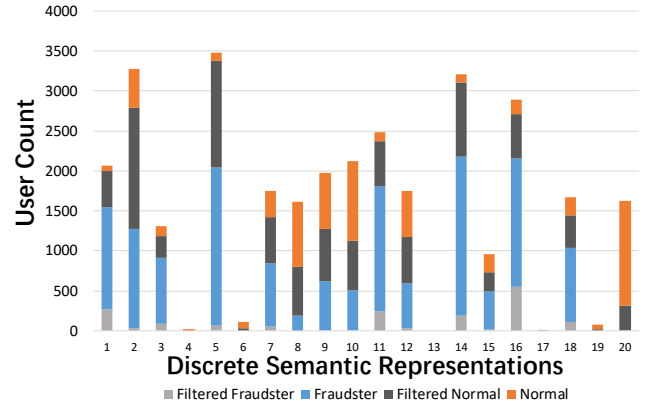


Figure 3: X-axis shows 20 bars and each bar is a specific value of the semantic representation (i.e. $d_n = 1, \dots, 20$). Y-axis indicates user count and orange, blue and gray blocks respectively illustrate normal users, fraudsters and outliers. Some normal users are not filtered out as Figure 5 described.

Figure 3 displays the user distribution after detection. Every data point is assigned according to $\max_g \tilde{\phi}_{ng}$. Each bar represents a possible value for d_n , and its height indicates the user count. We color the bars according to the ground truth of each data point, and we discuss the bars from top to bottom. The top layer (orange blocks) comprises normal users, and the third layer (blue blocks) comprises fraudsters. We filter out the outliers according to Section 3.2.3 and group them into the second and fourth layers (gray blocks).

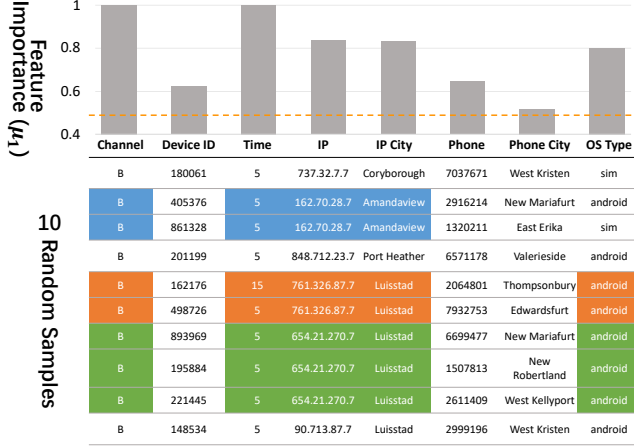


Figure 4: Visualization of the representation $d_n = 1$ from Figure 3. We randomly sample 10 data points with $d_n = 1$ and plot the feature importance (i.e. parameter μ , as it indicates how synchronized the feature is). The dashed line $\mu_{1,\cdot} = 0.5$ indicates neither the sparsity and randomness patterns overpower the other. We highlight the synchronized behaviors of the fraudsters in these samples with different background colors. Note that the three fraud groups (three different background color) share the same synchronized features, which are important features suggested by high μ values (much greater than 0.5).

We can see that most of the normal users are filtered out of the fraudulent groups (e.g., group 1, 2, and 5). However, there are still many normal users in these bars. As such, for our next step, we investigate one of these bars.

We randomly sample 10 data points from the first bar (i.e. setting $d_n = 1$), and plot the learned feature importance parameter (μ_1) in Figure 4. We can observe the overt synchronized behaviors of these records, which are highlighted as colored blocks. The parameter $\mu_{1,\cdot}$ for these features is relatively high (over 0.8), which indicates that these features are highly synchronized. We can see that in the *channel* and *time* features, the parameter is almost 1, as nearly all of the records with $d_n = 1$ have nearly the same value. The $\mu_{1,IP}$ value for feature *IP* is a little smaller than 1 because there are several popular IP addresses. In features like *device ID*, *phone* and *phone city*, the data points are less synchronized. As a result, the parameter $\mu_{1,\cdot}$ for these features is closer to 0.5, which signifies that the degrees of synchronization and randomness are similar.

We further investigate the IP feature with $d_n = 1$ to see the probability mass of those possible IP addresses ($\alpha_{1,IP,\cdot}$) as well as the number of data points that use these IP addresses. The result is plotted in Figure 5. The bars are again colored according to the ground truth labels of the data points to demonstrate user identities. We can see that the sampled IPs in Figure 4 have high probability mass, which means the model correctly captures these clustering patterns. Interestingly, we find some small groups of normal users that share the same IP addresses. This explains why some normal users remain in the bars in Figure 3. The largest group consists of 28

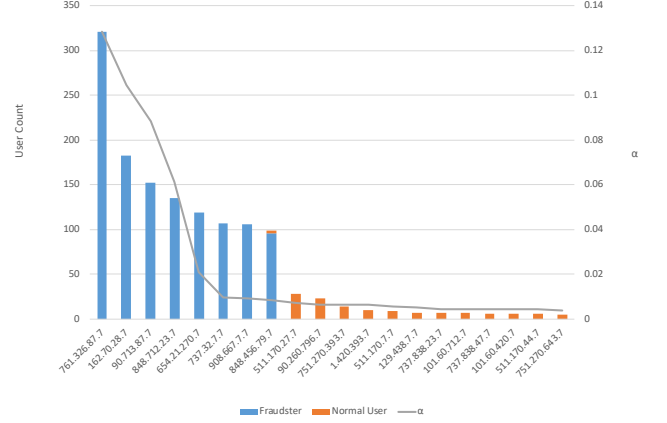


Figure 5: Investigate IP address distribution when setting $d_n = 1$. X-axis represents IP address and sorted w.r.t. the parameter α . Y-axis is the user count and we color bars according to ground truth and many IP addresses shared by groups of normal users.

users that share the IP address 511.170.27.7. Such synchronization on a single IP address is suspicious, and we should pay extra attention to these groups in practice. Note that $\alpha_{1,IP}$ decreases faster than the bars since the bars correspond to the likelihood in the objective function, and the regularizers force $\alpha_{1,IP}$ to be sparser than the likelihood. Together with the observations in Figure 4, we verify that FIRD can assign different fraudulent groups the same latent representation as long as the synchronized features of these groups are identical.

In a nutshell, this experiment shows that FIRD can detect reasonable fraud patterns and automatically infer the labels of each data point based on our prior knowledge. In addition to significant detection results, the probability representations learned by FIRD can also assist us in analyzing the fraud patterns.

4.2 Results on Anomaly Detection Benchmarks

FIRD can also serve as a general anomaly detection method by assigning $p(\ell|d_n = g) = 1$ for anomalies and $p(\ell|d_n = g) = 0$ for nominal data points in Eq. (10). Here we demonstrate the detection results of FIRD and comparison methods on benchmark anomaly detection datasets. We first describe the setup and then provide comparison results and analysis.

4.2.1 Experiment Setup. ODDS [29] organizes the benchmark datasets used in related works. These datasets come from different domains and are readily labeled by reorganizing some multi-class datasets, where the smallest class is chosen to be the anomaly [1, 3, 15, 21, 40]. We use the ROC-AUC score to evaluate the performance of the methods, and compares FIRD with the most promising methods³, namely the histogram-based outlier score (HBOS) [10], the isolation forests (IForest) [21], the one-class SVM (OCSVM) [9] and locally selective combination in parallel outlier ensembles (LSCP) [38].

³More comparison methods: <https://pyod.readthedocs.io/en/latest/benchmark.html>

Dataset	FIRD	HBOS	IForest	OCSVM	LSCP
cardio	0.949	0.843	0.924	0.938	0.901
musk	1.000	1.000	0.999	1.000	0.998
optdigits	1.000	0.865	0.714	0.500	-
satimage-2	0.998	0.977	0.993	0.997	0.9935
shuttle	0.990	0.986	0.997	0.992	0.5514
satellite	0.900	0.754	0.701	0.660	0.6015
ionosphere	0.946	0.5569	0.8529	0.8597	-
pendigits	0.972	0.9247	0.9435	0.931	0.8744
wbc	0.944	0.954	0.9325	0.9376	0.945

Table 2: Results on anomaly detection benchmark datasets. We evaluate the performance of identifying fraud users by ROC-AUC scores. The datasets *cardio*, *musk* and *optdigits* come from [3]. The datasets *satellite* and *ionosphere* come from [21]. The datasets *wbc* and *pendigits* come from [15]. The dataset *satimage-2* comes from [40], and *shuttle* comes from [1].

4.2.2 Comparison results. We demonstrate the comparison results in Table 2. FIRD displays competitive performance in most of these benchmarks, which implies that modeling both the synchronization and randomness benefits the detection. The state-of-the-art anomaly detection methods usually explore the heterogeneous statistical patterns of different features by resampling. They construct a sequence of datasets by resampling the features as well as data points and learn an independent outlier detector on each resampled dataset [2]. They expect that among these samples, the learner may luckily drop the non-informative features and suppress the fraction of outliers so that the detection performance obtained from the ensemble of these detectors is superior to that of learning one detector on the entire dataset.

However, such a sampling strategy consumes more computational resources since it requires training many independent detectors. The resampling cost to capture the correct feature subset also proliferates with the number of dimensions due to the exponentially many feature combinations. Besides, the sampling strategy restricts the learner to model the local feature patterns explicitly. In some features, only a small subset of the dataset display interesting patterns. If the learner discards these features, we cannot recognize these local patterns. These data points then become noise, which will further reduce performance. In comparison, FIRD jointly models the randomness and synchronization to provide better recognition results with a less computational cost.

4.3 Model Analysis

In this section, we analyze the effectiveness of FIRD on synchronized datasets, generated following the process in section 3.2. We first analyze the dimension G of the semantic variable $\mathbf{d}$. Then we analyze the weights λ of the regularizers. We finally show the running time of FIRD as the feature dimension M increases. For parameter analysis, we set $N = 20000$, $G = 20$, $M = 20$, and $D_m = 200$ for all features. For running time analysis, we use $N = 20000$, $G = 10$, $D_m = 30$ and M ranging from 10 to 100.

Since the task is similar to clustering, we apply FIRD, KMeans, and spectral clustering to the synchronized dataset to recover $\mathbf{d}$. The

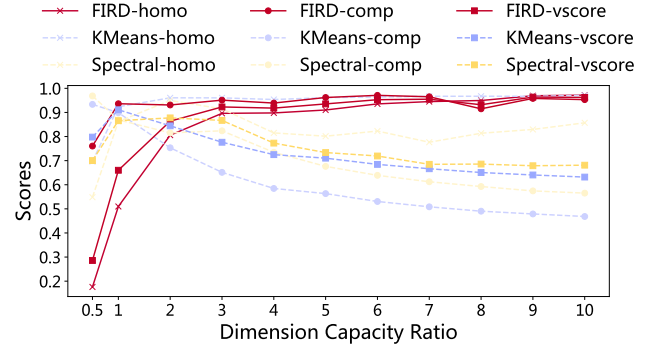


Figure 6: Performance under different dimension capacity ratio (DCR). DCR is calculated by G/G_{true} for FIRD and K/G_{true} for other two methods, where G_{true} is the ground truth latent dimension G used for data generation.

spectral clustering method [22] is a strong baseline for clustering performance, and Kmeans [14] is a fast baseline for clustering speed. We evaluate the clustering performance by three metrics:

- *Homogeneity* quantifies the pureness of the detected clusters. A high homogeneity score indicates the members of the cluster have almost the same d_n .
- A high *completeness* score means almost all the data points with $d_n = g$ are assigned to the same cluster.
- *V-score* is the harmonic mean between homogeneity and completeness.

All of the three metrics range from 0 to 1, larger values being desirable. A trivial strategy to achieve the best homogeneity score is to assign each data point as a cluster, while the completeness score will be 0. We can similarly achieve optimal completeness by assigning all data points to a single cluster, at the cost of zero homogeneity. Therefore, a good model is expected to optimize the three metrics simultaneously.

4.3.1 Different Choice of G . We first study the different choices of the dimension G of the latent discrete space. The parameter G controls the dimension of latent discrete space of $\mathbf{d}$, which is similar to the K of the KMeans and spectral clustering algorithm. For normalization, we demonstrate the result with different *dimension capacity ratio* (DCR) defined as the ratio of the dimension G to the ground truth dimension G_{true} , so the results are independent to G_{true} . We synchronize the dataset according to the generation process described in section 3.2 with $G_{true} = 10$. We then adopt $G = 5, 10, 20, \dots, 100$ for FIRD, and the same for K in KMeans and spectral clustering method to obtain DCR ranging from 0.5 to 10. Note that for KMeans and spectral clustering, we calculate the distance in the discrete space by applying a one-hot encoding preprocessing step to the vectors. The detection performance is demonstrated in Figure 6. It indicates that when the patterns are locally different for the clusters, conventional methods such as KMeans and spectral clustering require a perfect guess of G_{true} , i.e., DCR = 1, to achieve relatively high performance. As DCR increases, conventional methods tend to split the clusters into smaller ones due to the random noise in non-informative dimensions. In

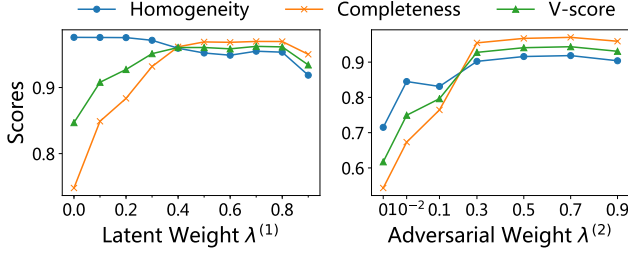


Figure 7: Performance under different regularization weights. FIRD is robust to the choice of the weights as long as the weights are not too small to cancel the effect of regularization.

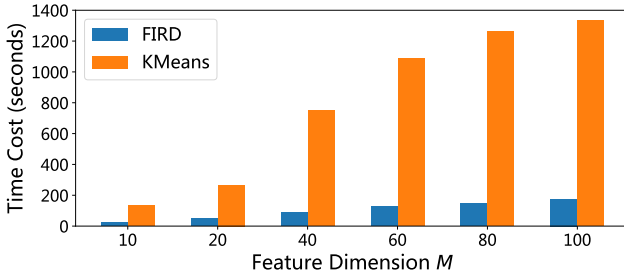


Figure 8: Running time of FIRD and KMeans as the feature dimension M increases. We implement FIRD with Cython and adopt KMeans implementation in the well-known Python package scikit-learn [25].

contrast, FIRD achieves high scores under all three metrics as long as the dimension capacity ratio is large enough, e.g., $G/G_{true} > 2$. As described in Section 3.2, FIRD automatically determines an appropriate G through the sparsity of π , so increasing the DCR does not affect its effectiveness.

4.3.2 Different Choice of λ . We introduced the normalized regularization weights $\lambda^{(1)}$ and $\lambda^{(2)}$ in section 3.2 and 3.2.5 to control the degree of sparsity or randomness in the parameters. The sparsity in π controlled by $\lambda^{(1)}$ enables the automatic determination of the appropriate G , and $\lambda^{(2)}$ reflects the difference between the adversarial distributions. We apply FIRD with different λ s to the synchronized dataset to study the effect, with results shown in Figure 7. We find that FIRD is robust to the choice of both $\lambda^{(1)}$ and $\lambda^{(2)}$ except when the weights are too small so that the learner no longer enjoys the modeling ability of the adversarial distributions.

4.3.3 Running Time Analysis. FIRD has linear computational cost w.r.t. the dataset scale, especially in high-dimensional spaces. We demonstrate the running time comparison of FIRD and KMeans in Figure 8. Here we adopt KMeans implementation in the well-known Python package scikit-learn [25]. Since KMeans calculates the distance between the one-hot encoded vectors, the computational cost increases exponentially with the feature dimension M . In contrast, FIRD decouples the features by the independence assumption, so it enjoys a linear growth in running time as M increases.

5 CONCLUSION

In this paper, we propose a novel unsupervised generative learning framework called FIRD to model heterogeneous statistical patterns in unlabeled datasets. FIRD utilizes the adversarial distributions with priors to capture such patterns. In discrete spaces, FIRD captures the synchronization and randomness patterns, which turns out quite useful on both fraud detection and general anomaly detection applications. The significant results on various datasets verify that modeling heterogeneous statistical patterns provides more generalizable representations and benefits various downstream applications. As future work, we expect FIRD to be effective in other applications that model the patterns other than the synchronization and randomness by adopting appropriate adversarial distributions.

ACKNOWLEDGMENTS

This research was supported by Alibaba Group through Alibaba Innovative Research Program.

A PROOF OF THEOREM 1

As the ELBO is concave w.r.t. all variables, it can be easily shown that the solution in Theorem 1 is optimal. To show the uniqueness, we prove that each adversarial component $p_k(x_{nm}|d_n, \theta)$ will fit the corresponding pattern $p_{k'}^*(x_{nm}|d_n)$. According to the definition

$$D_{KL}(p_{k'}^*(x_{nm}|d_n) \| p_k(x_{nm}|d_n, \theta)) = \mathbb{E}_{x \sim p_{k'}^*} [\log p_k(x_{nm}|d_n, \theta)] + H,$$

where H is the entropy of $p_{k'}^*(x_{nm}|d_n)$, the KL divergence assumption in Theorem 1 indicates for all possible d_n and $\forall j \neq k$,

$$\mathbb{E}_{x \sim p_{k'}^*} [p_k(x_{nm}|d_n, \theta)] > \mathbb{E}_{x \sim p_{k'}^*} [p_j(x_{nm}|d_n, \theta)]. \quad (12)$$

Since $\tilde{\mu}_{nmk} > 0$, multiplying Eq. (12) with $q(d_n)\tilde{\mu}_{nmk}$ and summing over m, k and d_n gives

$$\begin{aligned} & \lim_{N \rightarrow \infty} \frac{1}{N} ELBO(p_k \rightarrow p_{k'}^*) \\ &= \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n, d_n, m, k} q(d_n) \tilde{\mu}_{nmk} \log p_k(x_{nm}|d_n, \theta) + \text{const} \\ &\geq \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n, d_n, m, k} q(d_n) \tilde{\mu}_{nmk} \log p_j(x_{nm}|d_n, \theta) + \text{const} \\ &= \lim_{N \rightarrow \infty} \frac{1}{N} ELBO(p_j \rightarrow p_{k'}^*) \end{aligned}$$

which indicates that the optimal solution use $p_k(x_{nm}|d_n, \theta)$ to approximate the corresponding pattern $p_{k'}^*(x_{nm}|d_n)$. Using the EM algorithm gives the estimation of other parameters in Theorem 1, which completes the proof.

REFERENCES

- [1] Naoki Abe, Bianca Zadrozny, and John Langford. 2006. Outlier detection by active learning. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Philadelphia, PA, USA, August 20–23, 2006. 504–509. <https://doi.org/10.1145/1150402.1150459>
- [2] Charu C. Aggarwal. 2013. *Outlier Analysis*. Springer. <https://doi.org/10.1007/978-1-4614-6396-2>
- [3] Charu C. Aggarwal and Saket Sathe. 2015. Theoretical Foundations and Algorithms for Outlier Ensembles. *SIGKDD Explorations* 17, 1 (2015), 24–47. <https://doi.org/10.1145/2830544.2830549>
- [4] Salem Alelyani, Jiliang Tang, and Huan Liu. 2013. Feature Selection for Clustering: A Review. In *Data Clustering: Algorithms and Applications*. 29–60.

- [5] Andrew R. Barron, Jorma Rissanen, and Bin Yu. 1998. The Minimum Description Length Principle in Coding and Modeling. *IEEE Trans. Information Theory* 44, 6 (1998), 2743–2760. <https://doi.org/10.1109/18.720554>
- [6] Jianfei Chen, Jun Zhu, Yee Whye Teh, and Tong Zhang. 2018. Stochastic Expectation Maximization with Variance Reduction. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*. 7978–7988. <http://papers.nips.cc/paper/8021-stochastic-expectation-maximization-with-variance-reduction>
- [7] Constantinos Constantinopoulos, Michalis K. Titsias, and Aristidis Likas. 2006. Bayesian Feature and Model Selection for Gaussian Mixture Models. *IEEE Trans. Pattern Anal. Mach. Intell.* 28, 6 (2006), 1013–1018. <https://doi.org/10.1109/TPAMI.2006.111>
- [8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, 20–25 June 2009, Miami, Florida, USA. 248–255. <https://doi.org/10.1109/CVPRW.2009.5206848>
- [9] Franck Dufrenois and Jean-Charles Noyer. 2016. One class proximal support vector machines. *Pattern Recognition* 52 (2016), 96–112. <https://doi.org/10.1016/j.patcog.2015.09.036>
- [10] Markus Goldstein and Andreas Dengel. 2012. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012: Poster and Demo Track* (2012), 59–63.
- [11] Zengyou He, Xiaofei Xu, and Shengchun Deng. 2003. Discovering cluster-based local outliers. *Pattern Recognition Letters* 24, 9–10 (2003), 1641–1650. [https://doi.org/10.1016/S0167-8655\(03\)00003-5](https://doi.org/10.1016/S0167-8655(03)00003-5)
- [12] Meng Jiang, Alex Beutel, Peng Cui, Bryan Hooi, Shiqiang Yang, and Christos Faloutsos. 2015. A General Suspiciousness Metric for Dense Blocks in Multimodal Data. In *2015 IEEE International Conference on Data Mining, ICDM 2015, Atlantic City, NJ, USA, November 14–17, 2015*. 781–786. <https://doi.org/10.1109/ICDM.2015.61>
- [13] Alan Jovic, Karla Brkic, and Nikola Bogunovic. 2015. A review of feature selection methods with applications. In *38th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2015, Opatija, Croatia, May 25–29, 2015*. 1200–1205. <https://doi.org/10.1109/MIPRO.2015.7160458>
- [14] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. 2002. An Efficient k-Means Clustering Algorithm: Analysis and Implementation. *IEEE Trans. Pattern Anal. Mach. Intell.* 24, 7 (2002), 881–892. <https://doi.org/10.1109/TPAMI.2002.1017616>
- [15] Fabian Keller, Emmanuel Müller, and Klemens Böhm. 2012. HiCS: High Contrast Subspaces for Density-Based Outlier Ranking. In *IEEE 28th International Conference on Data Engineering (ICDE 2012)*, Washington, DC, USA (Arlington, Virginia), 1–5 April, 2012. 1037–1048. <https://doi.org/10.1109/ICDE.2012.88>
- [16] YongSeog Kim, W. Nick Street, and Filippo Menczer. 2002. Evolutionary model selection in unsupervised learning. *Intell. Data Anal.* 6, 6 (2002), 531–556. <http://content.iospress.com/articles/intelligent-data-analysis/ida00110>
- [17] Martin O. Larsson and Johan Ugander. 2011. A concave regularization technique for sparse mixture models. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12–14 December 2011, Granada, Spain*. 1890–1898. <http://papers.nips.cc/paper/4430-a-concave-regularization-technique-for-sparse-mixture-models>
- [18] Martin H. C. Law, Anil K. Jain, and Mário A. T. Figueiredo. 2002. Feature Selection in Mixture-Based Clustering. In *Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS 2002, December 9–14, 2002, Vancouver, British Columbia, Canada]*. 625–632. <http://papers.nips.cc/paper/2308-feature-selection-in-mixture-based-clustering>
- [19] Aleksandar Lazarevic and Vipin Kumar. 2005. Feature bagging for outlier detection. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, Illinois, USA, August 21–24, 2005*. 157–166. <https://doi.org/10.1145/1081870.1081891>
- [20] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *Computer Vision – ECCV 2014 – 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V*. 740–755. https://doi.org/10.1007/978-3-319-10602-1_48
- [21] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation Forest. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15–19, 2008, Pisa, Italy*. 413–422. <https://doi.org/10.1109/ICDM.2008.17>
- [22] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. 2001. On Spectral Clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3–8, 2001, Vancouver, British Columbia, Canada]*. 849–856. <http://papers.nips.cc/paper/2092-on-spectral-clustering-analysis-and-an-algorithm>
- [23] Søren Feodor Nielsen. 2000. The stochastic EM algorithm: estimation and asymptotic results. *Bernoulli* 6, 3 (2000), 457–489.
- [24] Girish Keshav Palshikar. 2002. The hidden truth-frauds and their control: A critical application for business intelligence. *Intelligent Enterprise* 5, 9 (2002), 46–51.
- [25] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2012. Scikit-learn: Machine Learning in Python. *CoRR* abs/1201.0490 (2012). [arXiv:1201.0490](http://arxiv.org/abs/1201.0490)
- [26] Adrian E. Raftery and Nema Dean. 2006. Variable selection for model-based clustering. *J. Amer. Statist. Assoc.* 101, 473 (2006), 168–178.
- [27] S Benson Edwin Raj and A Annie Portia. 2011. Analysis on credit card fraud detection methods. In *2011 International Conference on Computer, Communication and Electrical Technology (ICCCET)*. IEEE, 152–156.
- [28] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100, 000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1–4, 2016*. 2383–2392. <https://www.aclweb.org/anthology/D16-1264/>
- [29] Shebati Rayana. 2016. ODDS Library. <http://odds.cs.stonybrook.edu>
- [30] Kijung Shin, Bryan Hooi, and Christos Faloutsos. 2016. M-Zoom: Fast Dense-Block Detection in Tensors with Quality Guarantees. In *Machine Learning and Knowledge Discovery in Databases – European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19–23, 2016, Proceedings, Part I*. 264–280. https://doi.org/10.1007/978-3-319-46128-1_17
- [31] Kijung Shin, Bryan Hooi, Jisu Kim, and Christos Faloutsos. 2017. D-Cube: Dense-Block Detection in Terabyte-Scale Tensors. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6–10, 2017*. 681–689. <https://doi.org/10.1145/3018661.3018676>
- [32] Mei-Ling Shyu, Shu-Ching Chen, Kanoksri Sarinnapakorn, and LiWu Chang. 2003. A Novel Anomaly Detection Scheme Based on Principal Component Classifier. Technical Report. Department of Electrical And Computer Engineering, University of Miami.
- [33] Cláudia M. V. Silvestre, Margarida G. M. S. Cardoso, and Mário A. T. Figueiredo. 2015. Feature selection for clustering categorical data with an embedded modelling approach. *Expert Systems* 32, 3 (2015), 444–453. <https://doi.org/10.1111/exsy.12082>
- [34] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18–21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*. 1631–1642. <https://www.aclweb.org/anthology/D13-1170/>
- [35] Mahlet G Tadesse, Naijun Sha, and Marina Vannucci. 2005. Bayesian variable selection in clustering high-dimensional data. *J. Amer. Statist. Assoc.* 100, 470 (2005), 602–617.
- [36] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6–9, 2019*. <https://openreview.net/forum?id=rJ4km2R5t7>
- [37] Arthur J. White, Jason Wyse, and Thomas Brendan Murphy. 2016. Bayesian variable selection for latent class analysis using a collapsed Gibbs sampler. *Statistics and Computing* 26, 1–2 (2016), 511–527. <https://doi.org/10.1007/s11222-014-9542-5>
- [38] Yue Zhao, Zain Nasrullah, Maciej K. Hryniewicki, and Zheng Li. 2019. LSCP: Locally Selective Combination in Parallel Outlier Ensembles. In *Proceedings of the 2019 SIAM International Conference on Data Mining, SDM 2019, Calgary, Alberta, Canada, May 2–4, 2019*. 585–593. <https://doi.org/10.1137/1.9781611975673.66>
- [39] Zheng Zhao and Huan Liu. 2007. Spectral feature selection for supervised and unsupervised learning. In *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20–24, 2007*. 1151–1157. <https://doi.org/10.1145/1273496.1273641>
- [40] Arthur Zimek, Matthew Gaudet, Ricardo J. G. B. Campello, and Jörg Sander. 2013. Subsampling for efficient and effective unsupervised outlier detection ensembles. In *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11–14, 2013*. 428–436. <https://doi.org/10.1145/2487575.2487676>