Differentially Private High-Dimensional Data Publication via Markov Network

Fengqiong Wei, Wei Zhang(✉), Yunfang Chen, and Jingwen Zhao

School of Computer Science,
Nanjing University of Posts and Telecommunications, Nanjing, China
{1016041011,zhangw,chenyf,1017041019}@njupt.edu.cn

Abstract. Differentially private data publication has recently received considerable attention. However, it faces some challenges in differentially private high-dimensional data publication, such as the complex attribute relationships, the high computational complexity and data sparsity. Therefore, we propose *PrivMN*, a novel method to publish high-dimensional data with differential privacy guarantee. We first use the Markov model to represent the mutual relationships between attributes to solve the problem that the direction of relationship between variables cannot be determined in practical application. We then take advantage of approximate inference to calculate the joint distribution of high-dimensional data under differential privacy to figure out the computational and spatial complexity of accurate reasoning. Extensive experiments on real datasets demonstrate that our solution makes the published high-dimensional synthetic datasets more efficient under the guarantee of differential privacy.

Keywords: Differential privacy · High-dimensional data
Data publication · Markov network

1 Introduction

With the emergence of big data era, a large amount of user data is generated and accumulated, which becomes a new generation of resources to be urgently developed and utilized [1]. For instance, purchase records of online users is helpful for E-businesses to enhance the user experience and induce more consumption; patient information is helpful for doctors to improve the accuracy of diagnosis and level of medical services; population genetic database is helpful for scientists to predict disease and reduce the risk of illness. These data resources have such tremendous potential value. Therefore, how to make reasonable utilization is particularly important.

A vital issue of mining and using big data is privacy protection, which often involves the user's personal privacy leakage. If the data are shared directly or indirectly among the illegal person, it will make serious consequences [2]. Aiming at the problem of sharing and publishing private data, traditional solutions

R. Beyah et al. (Eds.): SecureComm 2018, LNICST 254, pp. 133–148, 2018.
https://doi.org/10.1007/978-3-030-01701-9_8

widely use anonymization technologies [3]. However, these anonymization technologies exist two obvious defects, cannot be quantified and cannot resist background attacks. In 2006, Dwork proposed the concept of differential privacy [4], which is a model of strict mathematical foundation and good robustness for privacy protection by adding controllable noise. Furthermore, it can resist the type of attacks in case of an attacker with specific background knowledge, and control the privacy leakage risk within acceptable limits. Differential privacy has been widely recognized in the industry and it has become a practical standard for privacy protection.

Differential privacy was originally designed to deal with simple relational data. However, with the development of big data, many high-dimensional and heterogeneous data appeared in practical applications. In the process of dealing with high-dimensional data, the biggest problem is the curse of dimensionality, that is, as the number of dimensions increases, the complexity and cost of analyzing and processing multi-dimensional data increases exponentially. Thus, one of the problems of high-dimensional data publishing is the sparsity of high-dimensional data. In consequence, it cannot guarantee utility by differential privacy since original data were covered by noise. Another problem, which is more prominent in high-dimensional data differential privacy publishing, is that the relationship between high-dimensional data is rather complicated and the change of single record will have a wider range of impact on the entire data, which results in the increase of data sensitivity. Therefore, for releasing high-dimensional data under differential privacy, it is important to reduce the data dimension and simplify the relationship between attributes to make the sensitivity controlled within a certain range.

To deal with the problem of high-dimensional data representation, researchers in the field of the Probabilistic Graphical Model [5] provide a new idea. They take advantage of the graph structure to represent the hidden relationship between various types of data and map all kinds of problems in applications onto the problem of calculating the probabilistic distribution of certain variables in the probabilistic model. The probabilistic graphical model provides the possibility of concise representation, efficient inference and learning various types of probability models. Therefore, it has been widely applied in many fields such as data processing and mining.

In this paper, considering the characteristics of high-dimensional data, we present a probabilistic graphical model for high dimensional data modeling and simplify the complex relationships between data onto the mutual relationship between variables. Specifically, we use Markov network to represent the probabilistic distribution of multiple random variables, consequently reducing the high-dimensional data dimension effectively and improving data utility. In addition, the inference algorithm in the probabilistic graphical model can effectively reduce computational complexity. Our contribution of this paper are as follows:

1. We propose the Markov network model to represent relationships between the variables without specifying directions of dependencies. The design of the potential function in undirected graph model is not constrained by the

probability distribution and more flexible. Meanwhile, it also avoids the constraint of global acyclic in directed graph model.

2. We develop the propagation-based approximate inference algorithm to deal with the NP-hard problem of exact inference algorithm as its computational complexity and spatial complexity grows exponentially. We specifically infer the distribution by the confidence-update propagation algorithm and this method can be applied to any structure network.

The remainder of the paper is organized as follows. The related work is presented in Sect. 2. Then, we describe some preliminaries in Sect. 3. The details of PrivMN are proposed in Sect. 4, followed by an extensive experimental evaluation in Sect. 5. Finally, a conclusion is depicted in Sect. 6.

2 Related Work

At present, the main research of differentially private data publication is how to guarantee the publishing accuracy of query result with the privacy budget. There are two kinds of applications, interactive data publishing and non-interactive data publishing.

The main question of interactive data publishing is how to answer as many data queries as possible with a limited privacy budget. In the early stage, Roth et al. [6] improved the *Laplace mechanism* proposed by Dwork et al. This method provides more inquiries under the same privacy budget. Gupta et al. [7] proposed a universal iterative dataset generation framework, which supports more queries as a whole. In general, the algorithm of interactive publishing method is relatively complicated, and the unknown of subsequent queries makes it have many limitations on query quantity and application mode.

The main problem of the non-interactive data publishing is how to design an efficient publishing algorithm to make it not only satisfy the differential privacy, but also has more utility. There are two main non-interactive data publishing strategies. One is adding noise to the original data and then optimize the data and publish the optimized result. Dwork [8] is an early representative method, which combines with *Laplace mechanism* to publish an equal-width histogram under differential privacy guarantee. However, one of the problems of histogram releasing is the consistency of the range query results. Therefore, many researchers propose some techniques to improve the availability and accuracy of the published equal-width histograms. For example, the post-processing method proposed by Hay et al. [9] makes the result of the publication guarantee the consistency under the condition of differential privacy, which not only satisfies the query accuracy but also reduces the noise addition.

However, the privacy cost of the above releasing strategy is relatively high. Therefore, another strategy is generally adopted, that is, convert or compress the original data first and then add noise to the processed data. For instance, Xiao et al. [10] first propose a multi-dimensional histogram distribution method DPCube that effectively reduces the query error. The wavelet transform method proposed by Xiao et al. [11] performs wavelet transform on the data before adding noise, which improves the accuracy of counting query to a certain extent. Barak et al. [12] propose the method of Fourier transform contingency table, which achieves the non-redundant encoding of marginal frequency. Meanwhile, the addition of the noise in the Fourier domain will not undermine the consistency between the edge frequencies.

When it comes to dealing with the problem of differential privacy protection for high-dimensional data, a basic idea is to propose an effective variable selection method to reduce the dimension to a reasonable degree (dimensionality reduction) on the premise of losing less information and then process the low-dimensional data. For example, Qardaji et al. [13] evenly divide two-dimensional spatial data onto equal-width cells and then add noise to each cell. Chen et al. [14] use a classification tree to generalize the high-dimensional dataset and finally publish noise counts. The *PriView* method proposed by Qardaji et al. [15] uses the cover design method of combination principle to select views, which decomposes the high-dimensional data onto the low-dimensional views, and then adds the noises to form the low-dimensional noisy marginal table, and finally uses the maximum entropy optimization algorithm to reconstruct the k-attribute marginal table for data publishing. Due to the increasing perturbation errors and computation complexity, Xu et al. [16] propose *DPPro* that publishes high-dimensional data via random projection to maximize utility while guaranteeing privacy. Ren et al. [17] identify correlations and joint distributions among multiple attributes to reduce the dimensionality of crowdsourced data, which achieves both efficiency and effectiveness.

Some attempts on differentially private data publishing have been made in the field of the probabilistic graphical model. Since Pearl [18] and Lauritzen [19] first introduced the concept of the graphical model into the field of artificial intelligence and statistical learning in the late 1980s, the graphical model has been rapidly applied to many fields. Zhang et al. [20] propose the *PrivBayes* method that uses the Bayesian network of the digraph model to represent the relationship between data attributes and combine a series of low-dimensional noise conditional probability tables by the chain rule of the Bayesian network to form a joint distribution for data publishing. Based on *PrivBayes*, Su et al. [21] present *DP-SUBN*, which develops a non-overlapping covering design (NOCD) method for generating all 2-way marginals of a given set of attributes to improve the fitness of the Bayesian network and reduce the communication cost. In addition, Xiao et al. [22] propose another scheme, which mainly uses attribute dependence graph to form attribute clusters, then adds noise to form low-dimensional noise marginal table, and finally publishing by sampling.

Different from the above solutions, we focus on the mutual relationship between multiple attributes, as well as the computational complexity and spatial complexity. To solve these problems, *PrivMN* uses the method of high-dimensional contingency table data publication and provides an approximate distribution of the original dataset based on the inference theory of probabilistic graphical model.

3 Preliminaries

3.1 Differential Privacy

Basic Definition. For a finite domain Z, $z \in$ Z is the element in Z. The dataset D is consist of z sampled from Z, its sample size is n and the number of attributes is dimension d.

Let datasets D and D' have the same attribute structure. The difference between them is denoted as $D \Delta D'$ and $\mid D \Delta D' \mid$ indicates the number of records in $D \Delta D'$. If $\mid D \Delta D' \mid = 1$, D and D' are called adjacent datasets.

Definition 1. *ϵ-Differential privacy [23]. A randomized algorithm M satisfies ϵ-Differential privacy, if for any two neighboring databases D and D', and for any $o \subseteq$ Range(M), $Pr[M(D) \in o] \leq \exp(\epsilon) \cdot Pr[M(D') \in o]$. Where the probability Pr[·] is taken over M's randomness and is the risk of privacy leakage. The parameter ϵ is privacy protection budget.*

From Definition 1, we can see that the privacy budget ϵ is used to control algorithm M to obtain same output probability ratio of two neighboring datasets, which reflects the level of privacy protection in fact. The smaller the value of ϵ, the higher the level of privacy protection. When ϵ equals 0, the protection level reaches the highest. At this time, the algorithm will output two identical probability distribution results for any neighboring dataset, but these results will not have any available information for a user.

Global Sensitivity. Differentially private protection can be achieved by adding an appropriate amount of interference noise to the return values of query function. Too much noise will affect the availability of the output, while too little will not provide enough security. The size of the noise is generally controlled by global sensitivity.

Definition 2. *Sensitivity [4]. Let f be a function that maps a dataset into a fixed-size vector of real numbers (i.e. $D \rightarrow R^d$). For two any neighboring databases D and D', the sensitivity of f is defined as $GS_f = \max_{D,D'} \| f(D) - f(D') \|_p$. Where p denotes L_p norm used to measure Δf, and we usually use L_1 norm.*

Noisy Mechanism. In practice, we usually add noise to algorithms to achieve differential privacy. In this paper, we rely on two best known and widely used, namely *Laplace mechanism* [8] and *exponential mechanism* [24]. The *Laplace mechanism* is suitable for numerical datasets, while the *exponential mechanism* is suitable for non-numerical datasets.

Laplace Mechanism. Laplace mechanism realizes the differential privacy by adding random noises that obey Laplace distribution to perturb the exact query result.

Theorem 1. *For any function $f : D \rightarrow R^d$, the mechanism M, $M(D) = f(D) + Y$, satisfies ϵ-Differential privacy, where $Y \sim Lap(\frac{\Delta f}{\epsilon})$ is i.i.d. Laplace variable with scale parameter $\frac{\Delta f}{\epsilon}$. The greater the sensitivity of algorithm M, the more amount of noise added.*

Exponential Mechanism. If the output is not numeric, we need to use availability function to evaluate the output. Let the output domain of query function is *Range*, and each value $r \in Range$ in the domain is an entity object. Under the *exponential mechanism*, the function $q(D, r) \rightarrow R$ is the availability function of the output value r, which is used to evaluate the quality of r.

Theorem 2. *Let the input of random M is dataset D, and output is an entity object $r \in Range$. $q(D, r)$ is availability function with its sensitivity, Δq. The mechanism M, $M(D, q) = \{r :| Pr[r \in Range] \propto \exp(\frac{\epsilon q(D,r)}{2\Delta q})\}$, satisfies ϵ-Differential privacy.*

3.2 Markov Network

Basic Conception. Markov Random Field (MRF) is also known as Markov Network. In general, the Markov Network is a complete joint probability distribution model for a group of random variables X which have Markov property [27], and *ISing Mode* is one of the earliest Markov Networks.

Definition 3. *Let $G = (V, E)$ be an undirected connection graph, where node $V_j \in V$ represents a random variable. If the node V_i and V_j in edge $(V_i, V_j) \in E$ satisfy the local Markov property:*

1. *The probability of each possible distribution is greater than 0.*
2. *The conditional probability distribution of an arbitrary node is only related to the value of its adjacent node (Locality).*

Then the network structure is called Markov Network, denoted as $\mathcal{H}$.

Conditional Independence. In the Markov network, there is a conclusion on the property of independence that if X_B 'splits' X_A and X_C, X_A and X_C are independent when X_B is given, and this property is also called Markov property.

Definition 4. *If a set of observed variables Z is given, there is no path between any two nodes $x \in X$ and $y \in Y$, then we call node set Z separates x and y in Markov network $\mathcal{H}$ and denoted as $sep_{\mathcal{H}}(X; Y \mid Z)$. The global independence associated with $\mathcal{H}$ is defined as: $I(\mathcal{H}) = \{X \perp Y \mid Z\} : sep_{\mathcal{H}}(X; Y \mid Z)$.*

Joint Probability Distribution

Definition 5. *According to Hammersley-Clifford Theorem [25, 26] and Local Markov Property, the joint probability distribution of Markov network is defined as:* $p(x) = \frac{1}{z}\prod_i \psi_i(x_i)$. $\psi_i(x_i)$ *is a non-negative real-valued function of* x_i*, which is usually called the potential function of a clique, and the variable* x_i *belongs to set* X. Z *is the normalization constant of partition function and its value is* $Z = \sum_x \prod_i \psi_i(x_i)$.

4 PrivMN Algorithm

4.1 PrivMN Overview

In this paper, we consider the following problem: Given a dataset D with d attributes, we want to generate a synthetic dataset that has approximate the joint distribution of original dataset D while satisfying differential privacy.

The method proposed in this paper includes the following four steps and the process of *PrivMN* is showed in Fig. 1:

1. Represent attributes relationship: we use a graphical model to represent the relationship between attributes and establish the Markov model.
2. Approximate inference: we infer approximately on the model based on the method of cluster graph confidence-propagation and obtain a series of low-dimensional marginal tables.
3. Generate noisy marginal: we add noise to the low-dimensional marginal table by *exponential mechanism* to form noisy marginal table.
4. Publishing synthetic datasets: we combine the noisy marginal tables and the Markov model to generate a synthetic dataset.

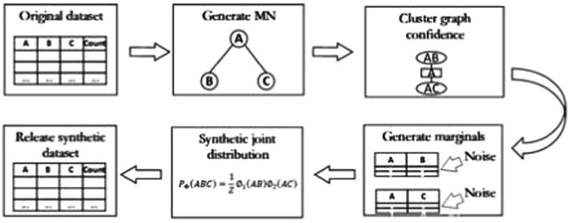


Fig. 1. The detail steps of *PrivMN*

4.2 Represent Attributes Relationship

As mentioned before, we use Markov network to represent the relationship between attributes. Firstly, we need to measure the relationship between attributes, there are many kinds of measures, such as chi-square test, mean-square contingency, Cramer's V coefficient, mutual Information and so on. In this paper, we choose mutual information to measure the correlation between two attributes. One reason is that mutual information is different from other correlation coefficients, that it is not limited to real-valued random variables and can express the degree of similarity generally. The other is not only for its small sensitivity but also for its capability of seizing the linear and non-linear correlations.

Given two attributes A_k and A_l, the mutual information $I(A_k, A_l)$ is defined as:

$$I(A_k, A_l) = \sum_{i=1}^{|\Omega_k|} \sum_{j=1}^{|\Omega_l|} p_{ij} \log \frac{p_{ij}}{p_{i \cdot} p_{\cdot j}} \tag{1}$$

where p_{ij} is the joint distribution of A_k and A_l. $p_{\cdot j} = \sum_j p_{ij}$ and $p_{\cdot j} = \sum_i p_{ij}$ is marginal distribution.

In this paper, we consider that A_k and A_l are independent if $I(A_k, A_l) \leq \theta_{kl}$ for some small threshold $\theta_{kl} > 0$. We choose Cramer's V coefficient as the threshold and Cramer's V coefficient is a method to calculate the correlation degree of between attributes in contingency table which attribute is greater than 2×2.

Cramer's V coefficient is calculated as follows:

$$\theta_{kl} = \sqrt{\frac{\chi^2}{n \min[(\mid \Omega_k \mid -1)(\mid \Omega_l \mid -1)]}} \tag{2}$$

where n is the size of a sample formed by two attributes, the domain of an attribute A_i is represented by Ω_i and its size is $\mid \Omega_i \mid$. χ^2 is the value of chi-square.

We present the process of establishing Markov network in Algorithm 1:

Algorithm 1. Establish Markov Network

```
Input: Dataset $D$ with attributes $A = \{A_1, A_2, \ldots A_d\}$
Input: Privacy parameter $\epsilon_1$
Output: Markov network $\mathcal{H}$
Initialize $H = (V, E)$ with $V = \{A_1, A_2, \ldots A_d\}$ and $E = \emptyset$;
$\eta = Lap(\frac{1}{\epsilon_1})$;
for each attribute pair $(A_k, A_l)$ do
    calculate $I(A_k, A_l)$;
    if $I(A_k, A_l) + \eta \geq \theta_{kl} + Lap(\frac{1}{\epsilon_1})$ then
        Add edge $(A_k, A_l)$ into $\mathcal{H}$;
return $\mathcal{H}$;
```

4.3 Approximate Inference

We have obtained the Markov network by Algorithm 1 which reveals attribute relations obviously. Then, we need to infer the model and the purpose of the inference is to achieve the marginal distribution and the conditional distribution of the given model. However, it is still complicated to obtain the required marginal distribution by inferring directly on the Markov network. Therefore, we need further clustering on the Markov network to reduce the computational complexity.

The cluster graph that we constructed in this step is a data structure, which provides a flowchart of the factor processing. Each node in the cluster graph is a cluster associated with a subset of the variables. The graph also contains undirected edges that connect non-empty intersection sets in the domain. Each edge between a pair of clusters C_i and C_j is relevant to a cut set $S_{i,j}$ that $S_{i,j} \subseteq C_i \cap C_j$. In addition, we make use of a simple structure called Bethe clustering graph, which can transform a general clustering graph into a clustering graph satisfying the confidence-propagation algorithm.

We obtain a series of clusters C_i and cut sets $S_{i,j}$ after clustering Markov network that satisfy the family-preserving of cluster graph: Each factor $\phi \in \Phi$ is related to a cluster graph C_i, expressed as $\alpha(\phi)$, and satisfy $Scope[\phi] \subseteq C_i$.

After obtaining the clustering graph, we ratiocinate in the clustering graph by the confidence-propagation algorithm in Algorithm 2. Confidence-propagation Algorithm of clustering Graph is an approximate calculation and iterative algorithm based on the undirected graph model. It updates the current probability distribution of the entire clustering graph by exchanging information between the nodes in the clustering graph. Moreover, it can solve probabilistic inference problems of the probabilistic graphical model and spread all information on parallel.

After several iterations, the confidence of all nodes is no longer changed. At this time, the clustering graph reaches the convergence state. Moreover, the marginal distribution of each cluster is the optimal solution. This cluster graph is called a cluster graph calibrated, that is, for each edge $(i-j)$ between connected clusters C_i and C_j in the cluster graph, there is

$$\mu_{i,j}(S_{i,j}) = \sum_{C_i - S_{i,j}} \beta_i(C_i) = \sum_{C_j - S_{i,j}} \beta_j(C_j) \tag{3}$$

Therefore, the confidence set $\mathcal{Q} = \{\beta_i : i \in vertex\, set\} \cup \{\mu_{i,j} : i - j \in edge\, set\}$ is a distribution similar to datasets. Where β_i denotes the confidence on C_i and $\mu_{i,j}$ represents the confidence on $S_{i,j}$.

We present the process of approximate inference in Algorithm 2:

Algorithm 2. Approximate Inference

```
Input: Markov network $\mathcal{H}$
Input: Factor set $\Phi$
Output: Confidence set $\mathcal{Q}$
Bethe cluster graph $\mathcal{U} \longleftarrow$ BehteGraphCreateAlgorithm($\mathcal{H}$);
confidence set $\mathcal{Q} \longleftarrow$ CGraph-SP-Calibrate($\mathcal{U}, \Phi$);
return $\mathcal{Q}$;
```

4.4 Generate Noisy Marginal

In this section, we use the *Laplace mechanism* to add noise to the marginal tables of each cluster to generate the noisy marginal tables and consequently realize the differential privacy protection for the attributes in the cluster.

Let the number of clusters be m. For each clusters marginal table, we add Laplace noise $Lap(\frac{m}{\epsilon_2})$ to each entry's count. Therefore, the privacy budget of a single cluster for privacy protection is $\frac{\epsilon_2}{m}$. According to the combinatorial property of the differential privacy protection algorithm, the differential privacy protection for different clusters in the same dataset provides the sum of all budgets. Therefore, the noisy marginal tables satisfy ϵ_2-differential privacy.

In order to reduce the error caused by adding noise and ensure the availability of noise-added data, we will post-process the noisy marginal tables. We cite the post-processing technique in [22] to ensure consistency even if the noisy marginal tables are of different sizes and attributes are not binary.

Let $A = C_1 \cap C_2 \cap \cdots C_m \neq \emptyset$, the public attribute of cluster group. We use T_{c_i} to denote C_i's noisy marginal table, $T_{c_i}[A]$ to denote A's marginal constructed from C_i and $T_{c_i}[A] \equiv T_{c_j}[A]$ to denote that two marginal tables are identical. We want to ensure $T_{c_i}[A] \equiv \cdots \equiv T_{c_m}[A]$, that is, all noisy marginal tables of an attribute are coincident.

We achieve this goal in two steps. Where a is a possible value in As domain and $T_A(a)$ is the count of a in As noisy marginal table.

1. Generate the approximate value of $T_A(a)$. The best estimate of $T_A(a)$ is the minimum noise variance. Therefore, we use inverse-variance weighting to obtain the variance of the weighted average as follows:

$$T_A(a) = \frac{\sum_{i=1}^{m} \frac{T_{c_i}(a)}{\sigma_i^2}}{\sum_i \frac{1}{\sigma_i^2}} \tag{4}$$

   where $\sigma_i^2 = \prod_{A_j \in (c_i \setminus A)} \mid \Omega_j \mid$ is proportional to the variance of $T_{c_i}[A](a)$.
2. Update all T_{c_i}s to be consistent with T_A:

$$T_{c_i}(e) \leftarrow T_{c_i}(e) + \frac{T_A(a) - T_{c_i}(a)}{\prod_{A_j \in (c_i \setminus A)} \mid \Omega_j \mid} \tag{5}$$

   where e is the a after the update.

To make all marginal tables consistent, we need to perform a series of mutual consistency steps.

In addition, in order to reduce the bias caused by rounding the negative noisy to 0 and assuring the accuracy, we turn negative counts into 0 while decreasing the counts for its neighbors to maintain overall count unchanged. Specifically, we choose a threshold θ that close to 0. The sum above the threshold is n and the sum below the threshold is k. For each count c above the threshold, we subtract $\mid k \mid * \frac{c}{n}$ as the last value of it, and the value below the threshold becomes 0.

4.5 Publishing Synthetic Datasets

Combining with the previously obtained clustering graph and the noisy marginal tables, we can calculate the joint distribution of attributes. Based on the joint probability calculation formula in Markov networks, the confidence set, and the noisy marginal tables, we can get the non-normalized distribution as follows:

$$\mathcal{P}_{\Phi}(\mathcal{H}) = \frac{\prod \beta_i(C_i)}{\prod \mu_{i,j}(S_{i,j})} \tag{6}$$

The normalization constant is usually obtained by the sum of all states, that is, $Z = \frac{\sum_{C_i} \prod \beta_i(C_i)}{\sum_{S_{i,j}} \prod \mu_{i,j}(S_{i,j})}$. Therefore, the joint distribution is calculated as follows:

$$P_{\Phi}(\mathcal{H}) = \frac{1}{Z}\mathcal{P}_{\Phi}(\mathcal{H}) = \frac{1}{Z}\frac{\prod \beta_i(C_i)}{\prod \mu_{i,j}(S_{i,j})} \tag{7}$$

However, directly sampling a synthetic dataset from the joint distribution is computationally prohibitive. Therefore, we use the clustering graph and the noisy marginal tables to generate a synthetic dataset. Specifically, the steps are as follows: 1. Randomly select a cluster in the cluster graph and sample its attributes from its noisy marginal distribution. 2. Continuously sample other attributes in the cliques adjacent to the cliques, that is, they share a common separator, and repeat the above operation. 3. Terminate this process until all the attributes have been sampled.

After the sampling, we calculate the joint distribution by using the joint probability calculation formula given earlier. Thus, we obtain the required joint distribution, which satisfies the differential privacy protection of the complete dataset.

In the four steps of *PrivMN*, only the first and third steps require access to the original dataset, so we divide the total privacy budget ϵ into two portions with ϵ_1 being used for the first step and ϵ_2 for the third step by the composition property [8,28]. Therefore, the first and third steps are ϵ_1- and ϵ_2-differential privacy respectively, and *PrivMN* satisfies -differential privacy as a whole, where $\epsilon = \epsilon_1 + \epsilon_2$.

5 Evaluation

We make use of three standard real datasets (both binary and non-binary) in our experiments. For binary datasets, we choose **Retail** referred from [22]. **Retail** is a retail market basket dataset, where each record consists of the distinct items purchased in a shopping visit. We preprocess **Retail** to include 50 binary attributes and its domain size is 2^{50}. For non-binary datasets, we use the same datasets used in [20]. **Adult** contains census data from 1994 US census. There are 15 non-binary attributes in it and its domain size is about 2^{52}. **TPC-E** contains information of 'Trade', 'Security', 'Security status' and 'Trade type' tables in the **TPC-E** benchmark. It consists of 24 non-binary attributes and its domain size is about 2^{77}.

We evaluate the *PrivMN* in two aspects: One is the construction of marginal table, which used to measure the accuracy of methods. The other is to train multiple SVM classifiers on the same dataset to predict attributes. We first generate synthetic datasets and then use these datasets to build SVM classifiers. The correct rate or error rate is the judgment of all data, which is the overall evaluation of the classifier and suitable for the evaluation of the experiment. Therefore, we use the error rate to measure the performance of the classifier and the property of the algorithm.

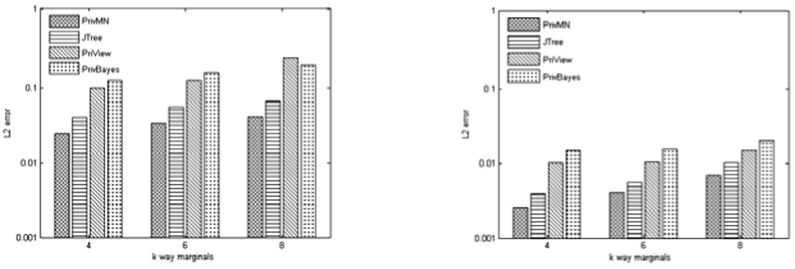
Since *PriView* [15] only works for binary datasets and cannot generate synthetic datasets for SVM classification, for binary datasets we only report the results on marginal tables. Due to L_2 error and Jensen-Shannon divergence are similar, we use the same evaluation scheme used in *PriView*, that is, we plot the average L_2 error where privacy budget $\epsilon \in \{0.1, 1.0\}$ and generate 200 random k-way marginal tables for each $k \in \{4, 6, 8\}$.

For non-binary datasets, when k is relatively large, a k-way marginal table is normally very sparse and the evaluation scheme used in binary datasets may be significantly biased. Therefore, we choose to follow the same methodology used in *PrivBayes* [20]. We generate all 2-way and 3-way marginal tables and perform the average total variation distance between the original datasets and the noisy datasets. In addition, we use the same method used in *PrivBayes* to test the classification results with SVM classifiers. We report the results on Adult, which is the most widely used benchmark dataset for SVM classification analysis. We train SVM classifiers on Adult to predict where an individual (1) is a male, (2) holds a post-secondary degree, (3) has salary $> 50k$ per year, and (4) has never married. We evaluate each classification task with privacy budget $\epsilon \in \{0.2, 0.5, 0.8, 1.0\}$. Each task uses 80% of the datasets as the training set and the remaining 20% for prediction. We employ the misclassification rate as the performance metric.

5.1 Contrast on Binary Datasets

In the first part of experiments, we compare the accuracy of four algorithms on the binary dataset by assigning different privacy budgets. The results are presented in Fig. 2.

It can be seen that our method, *PrivMN*, is far superior to *PrivBayes* in most cases and has some advantages over *PriView*. In Fig. 2(a), *PriView*s L_2 error is higher than *PrivBayes* when $k = 8$. It means that *PriView* is not stable and there is a substantial decrease in the performance of the property with the amount of attributes increase. Although *PrivMN* is similar to *JTree*, the error of *PrivMN* is smaller than *JTree*. Our method still maintains certain advantages as attributes increase. In general, the advantage of *PrivMN* is more observable when $\epsilon = 0.1$, that is, when ϵ is small, it is still the overall optimal without excessive volatility. Therefore, we consider the synthetic dataset generated by



(a) Retail, $\epsilon = 0.1$

(b) Retail, $\epsilon = 1.0$

Fig. 2. L_2 error of k-way marginals on binary datasets

PrivMN can meet different analysis needs. In addition, *PrivMN* can be applied to non-binary datasets, which is of great significance for practical applications.

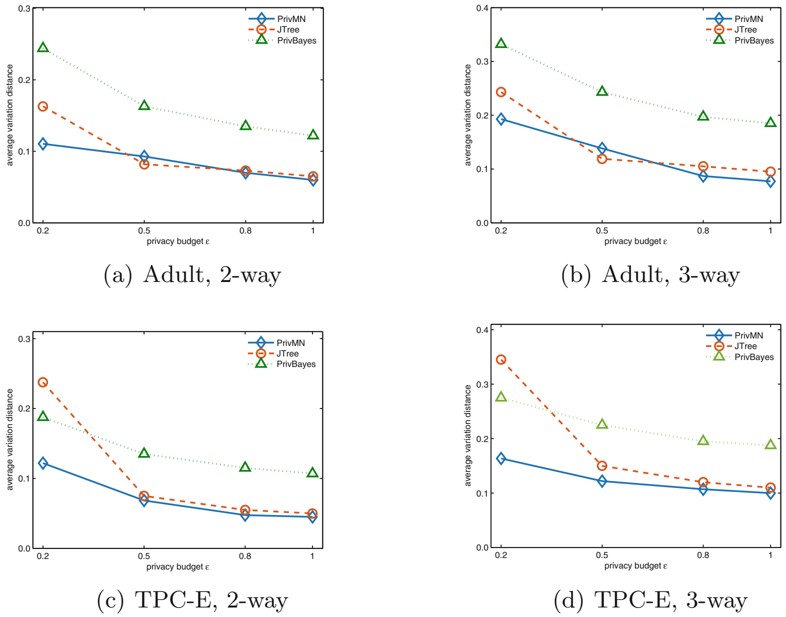
5.2 Contrast on Non-binary Datasets

***k*-Way Marginal Tables.** In the second part of the experiment, we compare the average total variation distance of three algorithms for varying privacy budgets on non-binary datasets and present the results in Fig. 3.

Since *PriView* cannot apply to non-binary datasets, we only compare the remaining three methods. It can be seen from the figure that the experimental results of *PrivMN* are far superior to *PrivBayes*. Under the condition of different datasets and different k-way marginal tables, the error of *JTree* is large when $\epsilon = 0.2$, and the overall change range is wide, especially in Fig. 3(c), (d). Although *PrivMN* makes more errors than *JTree* when $\epsilon = 0.5$ in Fig. 3(a), (b), it is relatively flat as a whole. With the gradual increase of the privacy budget, the added noise is less, and the average total variation distance is gradually reducing. Therefore, *PrivMN* is suitable for extensive datasets and is utility for many real-world applications.

SVM Classification. In the last part of experiments, we compare the misclassification rate to measure the performance of *PrivMN*, *JTree*, and *PrivBayes* on non-binary datasets. We report the results on Adult with different ϵ values in Fig. 4.

Non-Private is the misclassification rate of the original dataset, which is also the best experimental result we can achieve. In Fig. 4, *PrivMN* is far superior to *PrivBayes* in all cases. Compared with *JTree*, *PrivMN* decreases more slowly with different privacy budget, and the overall performance is better. In particular, *PrivMN* performs even better in Fig. 4(a), (b), (c). When $\epsilon = 0.2$ in Fig. 4(d), *PrivMN* has a slight fluctuation, but still within the acceptable range while *JTree* gets an obvious error. Although the property of the dataset generated by *PrivMN* is lower than that of the original dataset, it can satisfy the requirement of differential privacy and is superior to general methods. Therefore,



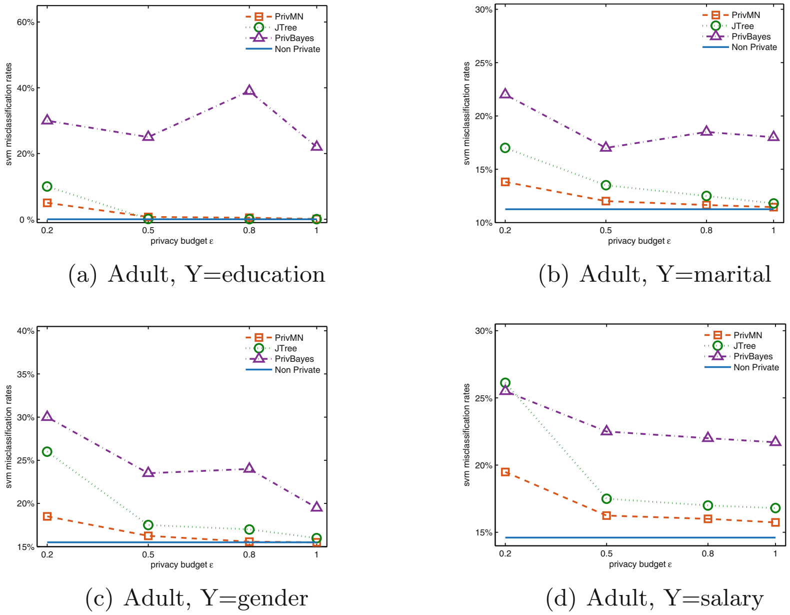
(a) Adult, 2-way

(b) Adult, 3-way

(c) TPC-E, 2-way

(d) TPC-E, 3-way

Fig. 3. Total variation distance of k-way marginal tables on non-binary datasets



(a) Adult, Y=education

(b) Adult, Y=marital

(c) Adult, Y=gender

(d) Adult, Y=salary

Fig. 4. SVM misclassification rates on non-binary datasets

PrivMN provides a generic data publishing solutions and it has certain practical significance.

6 Conclusion

Differentially private high-dimensional data publication is one of most challenging research issues and an important problem to be solved urgently. In this paper, we propose to use the Markov network model to represent the mutual relationships between attributes to solve the problem that the direction of relationship between variables cannot be determined in practical application. Moreover, we take advantage of approximate inference to calculate the joint distribution of high-dimensional data under differential privacy to figure out the computational and spatial complexity of accurate reasoning. Experiments on several real standard datasets demonstrate that *PrivMN* is significant in practice.

Acknowledgments. The authors would like to express their thanks to the anonymous reviewers for their constructive comments and suggestions. This work was supported by the National Natural Science Foundation of China under grants 61272422, 61672297.

References

1. The Economist: The worlds most valuable resource is no longer oil, but data, May 2017
2. Yu, S.: Big privacy: challenges and opportunities of privacy study in the age of big data. IEEE Access **2017**(4), 2751–2763 (2017)
3. Sweeney, L.: k-anonymity: a model for protecting privacy. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. **10**(5), 557–570 (2002)
4. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 1–12. Springer, Heidelberg (2006). https://doi.org/10.1007/11787006_1
5. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning. MIT Press, Cambridge (2009)
6. Roth, A., Roughgarden, T.: Interactive privacy via the median mechanism. In: Proceedings of the 42nd ACM Symposium on Theory of Computing, Cambridge, USA, pp. 765–774 (2010)
7. Gupta, A., Ligett, K., McSherry, F., et al.: Differentially private approximation algorithms (2009)
8. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_14
9. Hay, M., Rastogi, V., Miklau, G., et al.: Boosting the accuracy of differentially private histograms through consistency. Proc. VLDB Endow. **3**(1–2), 1021–1032 (2010)
10. Xiao, Y., Gardner, J., Xiong, L.: DPCube: releasing differentially private data cubes for health information. In: IEEE International Conference on Data Engineering, pp. 1305–1308. IEEE Computer Society (2012)

11. Xiao, X., Wang, G., Gehrke, J.: Differential privacy via wavelet transforms. IEEE Trans. Knowl. Data Eng. **23**(8), 1200–1214 (2011)
12. Barak, B., Chaudhuri, K., Dwork, C., et al.: Privacy, accuracy, and consistency too: a holistic solution to contingency table release. In: Proceedings of the 26th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, Beijing, China, pp. 273–282 (2007)
13. Qardaji, W., Yang, W., Li, N.: Differentially private grids for geospatial data. In: IEEE International Conference on Data Engineering, pp. 757–768. IEEE Computer Society (2013)
14. Chen, R., Mohammed, N., Fung, B.C.M., Desai, B.C., Xiong, L.: Publishing set-valued data via differential privacy. PVLDB **4**(11), 1087–1098 (2011)
15. Qardaji, W., Yang, W., Li, N.: Priview: practical differentially private release of marginal contingency tables. In: SIGMOD (2014)
16. Xu, C., Ren, J., Zhang, Y., et al.: DPPro: differentially private high-dimensional data release via random projection. IEEE Trans. Inf. Forensics & Secur. **12**(12), 3081–3093 (2017)
17. Ren, X., Yu, C.M., Yu, W., et al.: LoPub: high-dimensional crowdsourced data publication with local differential privacy. IEEE Trans. Inf. Forensics & Secur. **13**(9), 2151–2166 (2016)
18. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausble Inference. Morgan Kaufmann Publishers, Burlington (1988)
19. Lauritzen, S.L., Spiegelhalter, D.J.: Local computations with probabilities on graphical structures and their application to expert systems. J. R. Stat. Soc. Ser. B (Methodol.) **50**(2), 157–224 (1988)
20. Zhang, J., Cormode, G., Procopiuc, C.M., Srivastava, D., Xiao, X.: Privbayes: private data release via Bayesian networks. In: SIGMOD (2014)
21. Su, S., Tang, P., Cheng, X., et al.: Differentially private multi-party high-dimensional data publishing. In: IEEE International Conference on Data Engineering, pp. 205–216. IEEE (2016)
22. Chen, R., Xiao, Q., Zhang, Y., et al.: Differentially private high-dimensional data publication via sampling-based inference. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM (2015)
23. Dwork, C.: A firm foundation for private data analysis. Commun. ACM **54**(1), 86–95 (2011)
24. Mcsherry, F., Talwar, K.: Mechanism design via differential privacy. In: Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, pp. 94–103 (2007)
25. Hammersley, J.M., Clifford, P.: Markov fields on finite graphs and lattices (1971)
26. Cliord, P.: Markov random fields in statistics. Disord. Phys. Syst. A **14**(1), 128–135 (1990)
27. Zhang, J., Xiao, X., Xie, X.: PrivTree: a differentially private algorithm for hierarchical decompositions. In: International Conference on Management of Data, pp. 155–170. ACM (2016)
28. Li, D., Zhang, W., Chen, Y.: Differentially private network data release via stochastic kronecker graph. In: Cellary, W., Mokbel, M.F., Wang, J., Wang, H., Zhou, R., Zhang, Y. (eds.) WISE 2016. LNCS, vol. 10042, pp. 290–297. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48743-4_23