Graphical Model Sketch

Branislav Kveton KVETON@ADOBE.COM
Hung Bui HUBUI@ADOBE.COM
Mohammad Ghavamzadeh GHAVAMZA@ADOBE.COM
Georgios Theocharous THEOCHAR@ADOBE.COM
Adobe Research, San Jose, CA

S. Muthukrishnan MUTHU@CS.RUTGERS.EDU
Department of Computer Science, Rutgers

Siqi Sun SIQI.SUN@TTIC.EDU
TTI, Chicago, IL

Abstract

Structured high-cardinality data arises in many domains and poses a major challenge for both modeling and inference, which is beyond current graphical model frameworks. We view these data as a stream $(x^{(t)})_{t=1}^n$ of n observations from an unknown distribution P, where $x^{(t)} \in [M]^K$ is a K-dimensional vector and M is the cardinality of its entries, which is very large. Suppose that the graphical model $\mathcal{G}$ of P is known, and let $\bar{P}$ be the maximum-likelihood estimate (MLE) of P from $(x^{(t)})_{t=1}^n$ conditioned on $\mathcal{G}$. In this work, we design and analyze algorithms that approximate $\bar{P}$ with $\hat{P}$, such that $\hat{P}(x) \approx \bar{P}(x)$ for any $x \in [M]^K$ with a high probability, and crucially in the space independent of M. The key idea of our approximations is to use the structure of $\mathcal{G}$ and approximately estimate its factors by "sketches". The sketches hash high-cardinality variables using random projections. Our approximations are computationally and space efficient, being independent of M. Our error bounds are multiplicative and provably improve upon those of the count-min (CM) sketch, a state-of-the-art approach to estimating the frequency of values in a stream, in a class of naive Bayes models. We evaluate our algorithms on synthetic and real-world problems, and report an order of magnitude improvements over the CM sketch.

1. Introduction

Structured high-cardinality data arises in many domains, and poses a major challenge for both inference and modeling. In online advertising, for instance, the high-cardinality variables are the location of the user, the web page where the user lands, and the products that the user purchases. A common goal is to infer correlations among such variables. Graphical models (Jensen, 1996) are used extensively and successfully for modeling multivariate data. Unfortunately, when the cardinality of random variables is high, graphical models are expensive to store and infer with. As an example, a graphical model over two variables with $M = 10^5$ values each already consumes $M^2 = 10^{10}$ space.

A *sketch* is a data structure that summarizes the stream of data such that two sketches of individual streams can be combined space efficiently into the sketch of the combined stream (Muthukrishnan, 2005). Numerous problems can be solved by surprisingly simple sketches in a small space, such as estimating the frequency of values in the stream (Misra & Gries, 1982; Charikar et al., 2004; Cormode & Muthukrishnan, 2005a), finding heavy hitters (Cormode & Muthukrishnan, 2005b), estimating the number of unique values (Flajolet & Martin, 1985; Flajolet et al., 2007), or even approximating low-rank matrices (Liberty, 2013; Woodruff, 2014). In this work, we show how to sketch a graphical model in a small space.

Let $(x^{(t)})_{t=1}^n$ be a stream of n observations from distribution P, where $x^{(t)} \in [M]^K$ is a K-dimensional vector and P factorizes according to a known graphical model $\mathcal{G}$. Let $\bar{P}$ be the maximum-likelihood estimate (MLE) of P from $(x^{(t)})_{t=1}^n$ conditioned on graphical model $\mathcal{G}$. Our goal is to approximate $\bar{P}$ with $\hat{P}$ such that $\hat{P}(x) \approx \bar{P}(x)$ for any $x \in [M]^K$ and with a high probability, in the space independent of M, the cardinality of the variables in $\mathcal{G}$.

We make four contributions. First, we initiate the study of *graphical model sketches*, space-efficient approximations of factored distributions of high-cardinality variables from streams of data. Second, we propose and analyze three algorithms for building graphical model sketches. Our best solution, which we refer to as `GMFactorSketch`, approximates $\bar{P}$ in $O(Km \log(K/\delta))$ space such that $\hat{P}(x) \leq \bar{P}(x) \prod_{k=1}^{K}(1+\varepsilon_k)$ for any x and with at least $1-\delta$ probability, where $\hat{P}$ is the sketch of $\bar{P}$ and K is the number of variables in $\mathcal{G}$. Each error term ε_k is $O(1/m)$ and depends on the interaction of variable-parent pairs in model $\mathcal{G}$, where m is the number of hashing bins. Therefore, our approximations improve with the number of bins m and they are constant-factor multiplicative when $m = \Omega(K)$. Third, we prove that the error bound of `GMFactorSketch` is tighter than that of the count-min (CM) sketch (Cormode & Muthukrishnan, 2005a), a state-of-the-art approach to estimating the frequency of values in a stream, in a class of naive Bayes models. Finally, we evaluate our solutions on synthetic and real-world problems, and report an order of magnitude improvements in space over the CM sketch at the same quality of approximations.

The MLE is a fundamental approach to estimating the parameters of graphical models (Jensen, 1996). We propose and analyze space-efficient approximations to this estimation procedure with high-cardinality variables, which allow application of graphical models to many large-scale practical problems. Two recent papers studied related problems. McGregor & Vu (2015) proposed and analyzed a space-efficient algorithm that tests if the stream of data is consistent with a given graphical model. Belle et al. (2015) proposed and analyzed an approximate solver for *weighted model counting (WMC)* over discrete and continuous variables. In this paper, hashing is used to partition the space of counted models to speed up inference. The problem, as well as the context in which hashing is used, are different from those in this paper.

We denote $\{1, \dots, K\}$ by $[K]$. The cardinality of set A is $|A|$. We denote random variables by capital letters, such as X, and their values by small letters, such as x. We assume that $X = (X_1, \dots, X_K)$ is a K-dimensional variable; and we refer to its k-th component by X_k and its value by x_k.

2. Background

This section introduces two main ingredients of our solutions, count-min sketches and graphical models.

2.1. Count-Min Sketch

Let $(x^{(t)})_{t=1}^{n}$ be a stream of n observations from distribution P, where each $x^{(t)} \in [M]^K$ is a K-dimensional vector. Suppose that we want to estimate:

$$\tilde{P}(x) = \tfrac{1}{n} \textstyle\sum_{t=1}^{n} \mathbb{1}\left\{x = x^{(t)}\right\} , \qquad (1)$$

the empirical frequency of observing any x in $(x^{(t)})_{t=1}^{n}$. This problem can be solved exactly in $O(M^K)$ space, by counting all unique values in $(x^{(t)})_{t=1}^{n}$. This solution is not practical when K and M are large. Cormode & Muthukrishnan (2005a) proposed a data structure that can estimate $\tilde{P}(x)$ in the space which is independent of M^K. This data structure is known as the *count-min (CM) sketch*. The key idea is to maintain d hash tables with m bins each, $c \in \mathbb{N}^{d \times m}$. The hash tables are initialized with zeros. At time t, when $x^{(t)}$ is observed, the bins are updated as:

$$c(i, y) \leftarrow c(i, y) + \mathbb{1}\left\{y = h^i(x^{(t)})\right\}$$

for all $i \in [d]$ and $y \in [m]$, where $h^i : [M]^K \to [m]$ is the i-th *hash function*. The hash functions are random and *pairwise-independent*. The frequency $\tilde{P}(x)$ is estimated as:

$$P_{\text{CM}}(x) = \frac{1}{n} \min_{i \in [d]} c(i, h^i(x)) \, . \qquad (2)$$

Cormode & Muthukrishnan (2005a) show that $P_{\text{CM}}(x)$ approximates $\tilde{P}(x)$ for any $x \in [M]^K$, with at most ε error and at least $1-\delta$ probability, in $O((e/\varepsilon)\log(1/\delta))$ space. Perhaps surprisingly, the space is independent of M^K. We state this result more formally below.

Theorem 1. *Let $\tilde{P}$ be the distribution in (1) and P_{CM} be its CM sketch in (2). Let $d = \log(1/\delta)$ and $m = e/\varepsilon$. Then for any $x \in [M]^K$, $\tilde{P}(x) \leq P_{\text{CM}}(x) \leq \tilde{P}(x) + \varepsilon$ with probability of at least $1-\delta$. The space complexity of P_{CM} is $(e/\varepsilon)\log(1/\delta)$.*

The CM sketch is popular in practice because high-quality approximations, with at most ε error, require only $O(1/\varepsilon)$ space.[1] Other count sketches, such as that of Charikar et al. (2004), require $O(1/\varepsilon^2)$ space.

2.2. Bayesian Networks

Graphical models are a powerful tool for representing relationships of random variables (Wainwright & Jordan, 2008; Koller & Friedman, 2009) and have numerous applications in computer vision (Murphy et al., 2003), natural language processing (Lafferty et al., 2001) and bioinformatics (McDonald & Pereira, 2005). In this paper, we consider Bayesian networks (Jensen, 1996; Friedman et al., 1997), a directed graphical model.

A *Bayesian network* is a probabilistic graphical model that represents conditional independencies of random variables by a directed graph. Formally, it is a pair $(\mathcal{G}, \theta)$, where $\mathcal{G}$ is a directed graph and θ are the parameters of $\mathcal{G}$. The graph

[1] https://sites.google.com/site/countminsketch/

$\mathcal{G} = (V, E)$ is defined by its nodes $V = \{X_1, \dots, X_K\}$ and edges E. For simplicity of exposition, we assume in this work that each node in $\mathcal{G}$ has a single parent. So $\mathcal{G}$ is a *tree* and X_1 is its root. This assumption is not essential and can be easily relaxed (see Section 4.4). Under this assumption, the probability of $x = (x_1, \dots, x_K)$ factors as:

$$P(x) = P_1(x_1) \prod_{k=2}^{K} P_k(x_k \mid x_{\mathsf{pa}(k)}) \,,$$

where $\mathsf{pa}(k)$ is the *index of the parent variable* of X_k and we use shorthands:

$$\begin{aligned} P_k(i) &= P(X_k = i) \\ P_k(i, j) &= P(X_k = i, X_{\mathsf{pa}(k)} = j) \\ P_k(i \mid j) &= P_k(i, j) / P_{\mathsf{pa}(k)}(j) \,. \end{aligned}$$

Let $\mathrm{dom}\,(X_k) = M$ for all $k \in [K]$. Then our model $\mathcal{G}$ is parameterized by M *prior probabilities* $P_1(i)$, for any $i \in [M]$; and $(K-1)M^2$ *conditional probabilities* $P_k(i \mid j)$, for any $k \in [K] - \{1\}$ and $i, j \in [M]$.

Let $(x^{(t)})_{t=1}^n$ be n observations of X. Then the *maximum-likelihood estimate (MLE)* of P conditioned on $\mathcal{G}$:

$$\bar{\theta} = \arg\max_{\theta} \; P((x^{(t)})_{t=1}^n \mid \theta, \mathcal{G})$$

has a closed-form solution:

$$\bar{P}(x) = \bar{P}_1(x_1) \prod_{k=2}^{K} \bar{P}_k(x_k \mid x_{\mathsf{pa}(k)}) \,, \tag{3}$$

where we abbreviate $P(X = x \mid \bar{\theta}, \mathcal{G})$ as $\bar{P}(x)$, and:

$$\begin{aligned} \forall i \in [M] : \bar{P}_k(i) &= \frac{1}{n} \sum_{t=1}^{n} \mathbb{1}\left\{x_k^{(t)} = i\right\} \\ \forall i, j \in [M] : \bar{P}_k(i, j) &= \frac{1}{n} \sum_{t=1}^{n} \mathbb{1}\left\{x_k^{(t)} = i, x_{\mathsf{pa}(k)}^{(t)} = j\right\} \\ \forall i, j \in [M] : \bar{P}_k(i \mid j) &= \bar{P}_k(i, j) / \bar{P}_{\mathsf{pa}(k)}(j) \,. \end{aligned}$$

3. Model

Let $(x^{(t)})_{t=1}^n$ be a stream of n observations from distribution P, where each $x^{(t)} \in [M]^K$ is a K-dimensional vector. Our goal is to estimate $\bar{P}(x)$ in (3), the frequency of observing x as measured by the MLE of P from $(x^{(t)})_{t=1}^n$ conditioned on a known graphical model $\mathcal{G}$. This naturally generalizes the objective of the CM sketch in (1), which is the MLE of P from $(x^{(t)})_{t=1}^n$ without any assumptions on the structure of P. For simplicity of exposition, we assume that $\mathcal{G}$ is a tree (Section 2.2). Under this assumption, $\bar{P}$ can be represented exactly in $O(KM^2)$ space. In our problems, M is on the order of ten thousands or more. Therefore, it is impractical to represent $\bar{P}$ exactly.

The key idea in our solutions is to estimate a surrogate parameter $\hat{\theta}$. We estimate $\hat{\theta}$ on the same graphical model as $\bar{\theta}$. The difference is that $\hat{\theta}$ parameterizes a graphical model where the variables have a lower cardinality $m \ll M$. The new model consumes $O(Km^2)$ space, a significant reduction from $O(KM^2)$. We propose several new models, each of which guarantees that $P(x \mid \hat{\theta}, \mathcal{G}) \approx P(x \mid \bar{\theta}, \mathcal{G})$ for any $x \in [M]^K$ and any stream of observations $(x^{(t)})_{t=1}^n$ up to time n. In this sense, our guarantees are worst-case.

We write $\hat{P}(x)$ instead of $P(x \mid \hat{\theta}, \mathcal{G})$ and assume that $\hat{P}$ is factored as (3). Our algorithms guarantee that:

$$\bar{P}(x) \prod_{k=1}^{K} [1 - \varepsilon_k] \leq \hat{P}(x) \leq \bar{P}(x) \prod_{k=1}^{K} [1 + \varepsilon_k] \tag{4}$$

for any $x \in [M]^K$ with a high probability. Each error term ε_k is $O(1/m)$, where m is the number of hashing bins. Therefore, our approximations improve as the number of bins m increases. In particular, when m is chosen such that $\varepsilon_k \leq 1/K$ for any $k \in [K]$, we get that:

$$[2/(3e)]\bar{P}(x) \leq \hat{P}(x) \leq e\bar{P}(x) \tag{5}$$

for $K \geq 2$ because:

$$\begin{aligned} &\forall K \geq 1 : \textstyle\prod_{k=1}^{K} (1 + \varepsilon_k) \leq (1 + 1/K)^K \leq e \\ &\forall K \geq 2 : \textstyle\prod_{k=1}^{K} (1 - \varepsilon_k) \geq (1 - 1/K)^K \geq 2/(3e) \,. \end{aligned}$$

Therefore, $\hat{P}(x)$ is a constant-factor multiplicative approximation to $\bar{P}(x)$. Similarly to the CM sketch $P_{\text{CM}}(x)$, we do not require that $\hat{P}(x)$ sum up to 1.

4. Algorithms and Analysis

We propose three algorithms for computing $\hat{\theta}$, and analyze their space complexity and errors in (4). Each algorithm is presented and analyzed in a separate section. We compare the algorithms and discuss our results in Section 4.4.

All of our algorithms are based on a similar idea. We hash the values of each variable in graphical model $\mathcal{G}$, and each variable-parent pair, to m bins and potentially d times. We denote by h_k^i the i-th hash function of variable X_k, which also hashes the variable-parent pair $(X_k, X_{\mathsf{pa}(k)})$. The bins are a compact representation of $n\bar{P}_k(\cdot)$ and $n\bar{P}_k(\cdot, \cdot)$. We maintain hash tables for each variable and variable-parent pair in $\mathcal{G}$, and refer to them by c and $\bar{c}$, respectively. Our algorithms differ in how the hash tables are aggregated.

A *hash* is a tuple $h = (h_1, \dots, h_K)$ of K randomly drawn hash functions $h_k : \mathbb{N} \to [m]$, one for each variable in $\mathcal{G}$. We assume that any two hashes are pairwise-independent. Hashes h^i and h^j are *pairwise-independent* if h_k^i and h_k^j are pairwise-independent for any $k \in [K]$. Such hash functions can be computed very fast and stored in a very small space (Cormode & Muthukrishnan, 2005a).

Algorithm 1 `GMHash`: Hashed conditionals and priors.

Input: Point query $x = (x_1, \dots, x_K)$

$\hat{P}_1(x_1) \leftarrow \dfrac{c_1(h_1(x_1))}{n}$

for all $k = 2, \dots, K$ **do**

  $\hat{P}_k(x_k \mid x_{\mathsf{pa}(k)}) \leftarrow \dfrac{\bar{c}_k(h_k(x_k + M(x_{\mathsf{pa}(k)} - 1)))}{c_{\mathsf{pa}(k)}(h_{\mathsf{pa}(k)}(x_{\mathsf{pa}(k)}))}$

end for

$\hat{P}(x) \leftarrow \hat{P}_1(x_1) \prod_{k=2}^{K} \hat{P}_k(x_k \mid x_{\mathsf{pa}(k)})$

Output: Point answer $\hat{P}(x)$

4.1. Algorithm `GMHash`

The pseudocode of our first algorithm, `GMHash`, is shown in Algorithm 1. It approximates $\bar{P}(x)$ as the product of $K-1$ conditionals and a prior, one for each variable X_k. Each conditional is estimated as a ratio of two hashing bins:

$$\hat{P}_k(x_k \mid x_{\mathsf{pa}(k)}) = \frac{\bar{c}_k(h_k(x_k + M(x_{\mathsf{pa}(k)} - 1)))}{c_{\mathsf{pa}(k)}(h_{\mathsf{pa}(k)}(x_{\mathsf{pa}(k)}))},$$

where $\bar{c}_k(h_k(x_k + M(x_{\mathsf{pa}(k)} - 1)))$ counts the number of times that hash function h_k maps $(x_k^{(t)}, x_{\mathsf{pa}(k)}^{(t)})$ to the same bin as $(x_k, x_{\mathsf{pa}(k)})$ up to time n, and $c_k(h_k(x_k))$ counts the number of times that h_k maps $x_k^{(t)}$ to the same bin as x_k up to time n. Since $\mathrm{dom}\,(h_k) = \mathbb{N}$, we represent $(x_k, x_{\mathsf{pa}(k)})$ equivalently as $x_k + M(x_{\mathsf{pa}(k)} - 1)$. The prior $\bar{P}_1(x_1)$ is approximated as:

$$\hat{P}_1(x_1) = \frac{1}{n} c_1(h_1(x_1)) .$$

At time t, the hash tables are updated as follows. Let $x^{(t)}$ be the observation. Then for all $k \in [K], y \in [m]$:

$$c_k(y) \leftarrow c_k(y) + \mathbb{1}\left\{y = h_k(x_k^{(t)})\right\}$$
$$\bar{c}_k(y) \leftarrow \bar{c}_k(y) + \mathbb{1}\left\{y = h_k(x_k^{(t)} + M(x_{\mathsf{pa}(k)}^{(t)} - 1))\right\} .$$

This update takes $O(K)$ time. Now we show that $\hat{P}(x)$ is a good approximation of $\bar{P}(x)$.

Theorem 2. *Let $\bar{P}$ be the distribution in (3) and $\hat{P}$ be its estimator in Algorithm 1. Let h be a randomly drawn hash, where each hash function has m bins. Then for any x:*

$$\bar{P}(x) \prod_{k=1}^{K} (1 - \varepsilon_k) \leq \hat{P}(x) \leq \bar{P}(x) \prod_{k=1}^{K} (1 + \varepsilon_k)$$

holds with probability of at least $3/4$, where:

$$\varepsilon_1 = \frac{8K}{\bar{P}_1(x_1) m}$$
$$\forall k \in [K] - \{1\} : \varepsilon_k = \frac{8K}{\bar{P}_k(x_k, x_{\mathsf{pa}(k)}) m} .$$

Proof. A detailed proof is in Appendix A. The key idea is to show that for any $\varepsilon_1, \dots, \varepsilon_K > 0$, the number of bins m can be chosen such that:

$$|\hat{P}_k(x_k \mid x_{\mathsf{pa}(k)}) - \bar{P}_k(x_k \mid x_{\mathsf{pa}(k)})| > \varepsilon_k \tag{6}$$

is unlikely for any $k \in [K]$. In other words, our estimate of each conditional $\bar{P}_k(x_k \mid x_{\mathsf{pa}(k)})$ is sufficiently precise. By Lemma 1 in Appendix B, we get that the necessary conditions for event (6) are:

$$\frac{1}{n} c_{\mathsf{pa}(k)}(h_{\mathsf{pa}(k)}(x_{\mathsf{pa}(k)})) - \bar{P}_{\mathsf{pa}(k)}(x_{\mathsf{pa}(k)}) > \varepsilon_k \alpha_k$$
$$\frac{1}{n} \bar{c}_k(h_k(x_k + M(x_{\mathsf{pa}(k)} - 1))) - \bar{P}_k(x_k, x_{\mathsf{pa}(k)}) > \varepsilon_k \alpha_k ,$$

where $\alpha_k = \bar{P}_{\mathsf{pa}(k)}(x_{\mathsf{pa}(k)})$ is the frequency that $X_{\mathsf{pa}(k)} = x_{\mathsf{pa}(k)}$ in our stream. In summary, event (6) happens only if `GMHash` significantly overestimates either $\bar{P}_{\mathsf{pa}(k)}(x_{\mathsf{pa}(k)})$ or $\bar{P}_k(x_k, x_{\mathsf{pa}(k)})$. We bound the probability of the above events by Markov's inequality (Lemma 2 in Appendix B).

Now we select $m \geq 8 \sum_{k=1}^{K} (1/(\varepsilon_k \alpha_k))$ and get that none of the events in (6) happen with probability of at least $3/4$. Finally, we select appropriate $\varepsilon_1, \dots, \varepsilon_K$ and get our final error bound. ∎

Theorem 2 shows that $\hat{P}(x)$ is a multiplicative approximation to $\bar{P}(x)$. The approximation improves with the number of bins m because all error terms ε_k are $O(1/m)$. We note that the accuracy of the approximation depends on the frequency of interaction between the values in x. In particular, if $\bar{P}_k(x_k, x_{\mathsf{pa}(k)})$ is sufficiently large for all $k \in [K]$, which implies that the values of X_k and $X_{\mathsf{pa}(k)}$ co-occur frequently, the approximation is very good even for small m. More precisely, under the assumptions that:

$$m \geq \frac{1}{8K^2} \bar{P}_1(x_1)$$
$$\forall k \in [K] - \{1\} : m \geq \frac{1}{8K^2} \bar{P}_k(x_k, x_{\mathsf{pa}(k)}) ,$$

we get that $\varepsilon_k \leq 1/K$ for all $k \in [K]$; and the bounds in Theorem 2 reduce to (5) for $K \geq 2$.

`GMHash` stores $2K - 1$ hash tables, one for each variable X_k and one for each variable-parent pair $(X_k, X_{\mathsf{pa}(k)})$. All hash tables have m bins, and therefore `GMHash` consumes $O(Km)$ space.

4.2. Algorithm `GMSketch`

The pseudocode of our second algorithm, `GMSketch`, is in Algorithm 2. The algorithm approximates $\bar{P}(x)$ as the median of d probability estimates:

$$\hat{P}(x) = \text{median}_{i \in [d]} \hat{P}^i(x) \,.$$

Each $\hat{P}^i(x)$ is computed by one instance of `GMHash`, which is associated with the hash $h^i = (h_1^i, \dots, h_K^i)$. The hashes are random and pairwise-independent. At time t, the hash tables are updated as follows. Let $x^{(t)}$ be the observation. Then for all $k \in [K], i \in [d], y \in [m]$:

$$c_k(i, y) \leftarrow c_k(i, y) + \mathbb{1}\left\{y = h_k^i(x_k^{(t)})\right\} \qquad (7)$$
$$\bar{c}_k(i, y) \leftarrow \bar{c}_k(i, y) + \mathbb{1}\left\{y = h_k^i(x_k^{(t)} + M(x_{\mathsf{pa}(k)}^{(t)} - 1))\right\} \,.$$

This update takes $O(Kd)$ time. Now we show that $\hat{P}(x)$ is a good approximation of $\bar{P}(x)$.

Theorem 3. *Let $\bar{P}$ be the distribution in (3) and $\hat{P}$ be its estimator in Algorithm 2. Let $h^1, \dots, h^d$ be d random and pairwise-independent hashes, where each hash function has m bins. Then for any $d \geq 8 \log(1/\delta)$ and x:*

$$\bar{P}(x) \prod_{k=1}^K (1 - \varepsilon_k) \leq \hat{P}(x) \leq \bar{P}(x) \prod_{k=1}^K (1 + \varepsilon_k)$$

holds with probability of at least $1 - \delta$, where the error terms ε_k are defined as in Theorem 2.

Proof. A detailed proof is in Appendix A. The key idea is the so-called median trick. More precisely, if a real-valued random variable is in an interval with probability of at least $3/4$, the probability that the median of d i.i.d. samples of this variable is not in that interval is at most $\exp[-d/8]$ by Hoeffding's inequality. Now select $d = 8 \log(1/\delta)$ and we have that the median is not in the interval with probability of at most δ. ■

Similarly to Section 4.1, Theorem 3 shows that $\hat{P}(x)$ is a multiplicative approximation to $\bar{P}(x)$. The approximation improves with the number of bins m and depends on the frequency of interaction between the values in x.

The key benefit of maintaining d instances of `GMHash` is that we can get the same error guarantees as in Theorem 2 with probability of at least $1 - \delta$, in exchange for $\log(1/\delta)$ more space. Therefore, the space complexity of `GMSketch` is $O(Km \log(1/\delta))$.

4.3. Algorithm `GMFactorSketch`

Our final algorithm, `GMFactorSketch`, is in Algorithm 3. The algorithm approximates $\bar{P}(x)$ as the product of $K - 1$ conditionals and a prior, one for each variable X_k. Each conditional is estimated as a ratio of two CM sketches:

$$\hat{P}_k(x_k \mid x_{\mathsf{pa}(k)}) = \frac{\hat{P}_k(x_k, x_{\mathsf{pa}(k)})}{\hat{P}_{\mathsf{pa}(k)}(x_{\mathsf{pa}(k)})} \,,$$

where $\hat{P}_k(x_k, x_{\mathsf{pa}(k)})$ is the CM sketch of $\bar{P}_k(x_k, x_{\mathsf{pa}(k)})$, and $\hat{P}_k(x_k)$ is the CM sketch of $\bar{P}_k(x_k)$. The prior $\bar{P}_1(x_1)$ is approximated by its CM sketch $\hat{P}_1(x_1)$.

At time t, the hash tables are updated as in `GMSketch`, in (7). This update takes $O(Kd)$ time. The accuracy of the estimate $\hat{P}(x)$ is bounded below.

Theorem 4. *Let $\bar{P}$ be the distribution in (3) and $\hat{P}$ be its estimator in Algorithm 3. Let $h^1, \dots, h^d$ be d random and pairwise-independent hashes, where each hash function has m bins. Then for any $d \geq \log(2K/\delta)$ and x:*

$$\bar{P}(x) \prod_{k=1}^K (1 - \varepsilon_k) \leq \hat{P}(x) \leq \bar{P}(x) \prod_{k=1}^K (1 + \varepsilon_k)$$

holds with probability of at least $1 - \delta$, where:

$$\varepsilon_1 = \frac{e}{\bar{P}_1(x_1) m}$$
$$\forall k \in [K] - \{1\} : \varepsilon_k = \frac{e}{\bar{P}_k(x_k, x_{\mathsf{pa}(k)}) m} \,.$$

Proof. A detailed proof is in Appendix A. The outline of the proof is similar to that of Theorem 2. The difference is that we prove that event (6) is unlikely for any $k \in [K]$ by bounding the probabilities of the events:

$$\hat{P}_{\mathsf{pa}(k)}(x_{\mathsf{pa}(k)}) - \bar{P}_{\mathsf{pa}(k)}(x_{\mathsf{pa}(k)}) > \varepsilon_k \alpha_k$$
$$\hat{P}_k(x_k, x_{\mathsf{pa}(k)}) - \bar{P}_k(x_k, x_{\mathsf{pa}(k)}) > \varepsilon_k \alpha_k \,,$$

Algorithm 2 `GMSketch`: Median of d `GMHash` estimates.

Input: Point query $x = (x_1, \dots, x_K)$

for all $i = 1, \dots, d$ **do**
  $\hat{P}_1^i(x_1) \leftarrow \dfrac{c_1(i, h_1^i(x_1))}{n}$
  for all $k = 2, \dots, K$ **do**
    $\hat{P}_k^i(x_k \mid x_{\mathsf{pa}(k)}) \leftarrow \dfrac{\bar{c}_k(i, h_k^i(x_k + M(x_{\mathsf{pa}(k)} - 1)))}{c_{\mathsf{pa}(k)}(i, h_{\mathsf{pa}(k)}^i(x_{\mathsf{pa}(k)}))}$
  end for
  $\hat{P}^i(x) \leftarrow \hat{P}_1^i(x_1) \prod_{k=2}^K \hat{P}_k^i(x_k \mid x_{\mathsf{pa}(k)})$
end for
$\hat{P}(x) = \text{median}_{i \in [d]} \hat{P}^i(x)$

Output: Point answer $\hat{P}(x)$

Algorithm 3 `GMFactorSketch`: Count-min sketches of conditionals and priors.

Input: Point query $x = (x_1, \dots, x_K)$

// Count-min sketches for variables in $\mathcal{G}$
for all $k = 1, \dots, K$ **do**
  for all $i = 1, \dots, d$ **do**
    $\hat{P}_k^i(x_k) \leftarrow \dfrac{c_k(i, h_k^i(x_k))}{n}$
  end for
  $\hat{P}_k(x_k) \leftarrow \min_{i \in [d]} \hat{P}_k^i(x_k)$
end for

// Count-min sketches for variable-parent pairs in $\mathcal{G}$
for all $k = 2, \dots, K$ **do**
  for all $i = 1, \dots, d$ **do**
    $\hat{P}_k^i(x_k, x_{\mathsf{pa}(k)}) \leftarrow$
      $\dfrac{\bar{c}_k(i, h_k^i(x_k + M(x_{\mathsf{pa}(k)} - 1)))}{n}$
  end for
  $\hat{P}_k(x_k, x_{\mathsf{pa}(k)}) \leftarrow \min_{i \in [d]} \hat{P}_k^i(x_k, x_{\mathsf{pa}(k)})$
end for

for all $k = 2, \dots, K$ **do**
  $\hat{P}_k(x_k \mid x_{\mathsf{pa}(k)}) \leftarrow \dfrac{\hat{P}_k(x_k, x_{\mathsf{pa}(k)})}{\hat{P}_{\mathsf{pa}(k)}(x_{\mathsf{pa}(k)})}$
end for
$\hat{P}(x) \leftarrow \hat{P}_1(x_1) \prod_{k=2}^{K} \hat{P}_k(x_k \mid x_{\mathsf{pa}(k)})$

Output: Point answer $\hat{P}(x)$

where $\hat{P}_k(x_k, x_{\mathsf{pa}(k)})$ is the CM sketch of $\bar{P}_k(x_k, x_{\mathsf{pa}(k)})$, $\hat{P}_{\mathsf{pa}(k)}(x_{\mathsf{pa}(k)})$ represents the CM sketch of $\bar{P}_{\mathsf{pa}(k)}(x_{\mathsf{pa}(k)})$. The probabilities of the above events are bounded based on Cormode & Muthukrishnan (2005a). ■

As in Sections 4.1 and 4.2, Theorem 4 shows that $\hat{P}(x)$ is a multiplicative approximation to $\bar{P}(x)$. The approximation improves with the number of bins m and depends on the frequency of interaction between the values in x.

`GMFactorSketch` stores $O(\log(K/\delta))$ hashes, each with $2K - 1$ hash tables and m bins. Therefore, the space complexity of `GMFactorSketch` is $O(Km \log(K/\delta))$. The key difference with `GMHash` and `GMSketch` (Theorems 2 and 3) is that here the errors ε_k do not depend on K. Therefore, `GMFactorSketch` can achieve the same accuracy as `GMSketch` with K times less bins. If m is the number of bins in `GMSketch`, `GMFactorSketch` can achieve the same accuracy with m/K bins in $O(m \log(K/\delta))$ space, almost K times less space than in `GMSketch`.

4.4. Discussion

We propose three algorithms. `GMHash` approximates $\bar{P}(x)$ as the product of $K - 1$ conditionals and a prior, one for each variable X_k. Each conditional is estimated as a ratio of two hashing bins. `GMHash` guarantees (4) with probability of at least $3/4$ in $O(Km)$ space. This is a weak probabilistic guarantee in practice.

`GMSketch` approximates $\bar{P}(x)$ as the median of d probabilities, each of which is computed by an instance of `GMHash`. `GMSketch` guarantees (4) with probability of at least $1 - \delta$ in $O(Km \log(1/\delta))$ space. The dependence on δ is satisfactory. But the dependence on K is suboptimal.

Our last algorithm, `GMFactorSketch`, approximates $\bar{P}(x)$ as the product of $K - 1$ conditionals and a prior, one for each variable X_k. Each conditional is estimated as a ratio of two CM sketches. This algorithm guarantees (4) with probability of at least $1 - \delta$ in $O(Km \log(K/\delta))$ space. The key difference from `GMHash` and `GMSketch` is that its errors ε_k (Theorem 4) do not depend on K. As a result, for any $x \in [M]^K$, `GMFactorSketch` achieves the same quality of the approximation as the other two approaches in nearly K times less space.

The assumption that $\mathcal{G}$ is a tree (Section 3) is only for simplicity of exposition. Our algorithms and analysis generalize to the case where $X_{\mathsf{pa}(k)}$ is a vector of parent variables and $x_{\mathsf{pa}(k)}$ are their values. The only change is in how the tuples $(x_k, x_{\mathsf{pa}(k)})$ are hashed.

5. Comparison with the Count-Min Sketch

In general, the error bounds in Theorems 1 and 4 are hard to compare, because $\tilde{P}$ in (1) is a different estimator from $\bar{P}$ in (3). To compare the bounds, we make the assumption that $(x^{(t)})_{t=1}^n$ is a stream of observations such that $\bar{P} = \tilde{P}$. Such streams clearly exist. For instance, when $(x^{(t)})_{t=1}^n$ is drawn i.i.d. from P, both $\bar{P} \to P$ and $\tilde{P} \to P$ as $n \to \infty$ because both $\bar{P}$ and $\tilde{P}$ are consistent estimators of P. In the rest of this section, and without loss of generality, we assume that $\bar{P} = \tilde{P} = P$.

In general, `GMFactorSketch` is not guaranteed to improve over the CM sketch. As an example, consider a graphical model where the probability mass is concentrated in just a few points x. The CM sketch of these points would likely be a better approximation of $P(x)$ than `GMFactorSketch`. In the absence of such a sparsity structure, we believe that `GMFactorSketch` would approximate $P(x)$ better than the CM sketch for most queries x.

In this section, we construct a class of graphical models where the error bound of `GMFactorSketch` is tighter than that of the CM sketch. The model is a naive Bayes model with $K+1$ variables:

$$P(x) = P_1(x_1) \prod_{k=2}^{K+1} P_k(x_k \mid x_1) . \tag{8}$$

Variable X_1 is binary. For any $k \in [K+1] - \{1\}$, variable X_k takes values from $[M]$. For simplicity of exposition, we assume that the prior is $P_1(1) = P_1(2) = 0.5$. We fix x and define $C_k = P_k(x_k \mid x_1)$ for any $k \in [K+1] - \{1\}$.

Suppose that `GMFactorSketch` represents P_1 exactly, and therefore $\hat{P}_1 = P_1$. Then by Theorem 4:

$$\hat{P}(x) \leq \frac{1}{2}\left[\prod_{k=2}^{K+1} C_k\right]\left[\prod_{k=2}^{K+1}\left(1 + \frac{2e}{C_k m}\right)\right] \tag{9}$$

for any x with at least $1-\delta$ probability, where m is the number of bins in `GMFactorSketch` and we omit $1+\varepsilon_1$ from Theorem 4 because $\hat{P}_1 = P_1$. Our approximation consumes, up to logarithmic factors in K, $2Km\log(1/\delta)$ space. The CM sketch (Section 2.1) guarantees that:

$$\begin{aligned} P_{\text{CM}}(x) &\leq \frac{1}{2}\left[\prod_{k=2}^{K+1} C_k\right] + \frac{e}{m'} \\ &= \frac{1}{2}\left[\prod_{k=2}^{K+1} C_k\right]\left(1 + \frac{2e}{m'}\left[\prod_{k=2}^{K+1} \frac{1}{C_k}\right]\right) \end{aligned} \tag{10}$$

for any x with at least $1-\delta$ probability, where m' is the number of bins in the CM sketch. This approximation consumes $m'\log(1/\delta)$ space.

We want to show that the upper bound in (9) is tighter than that in (10) for any reasonable m. Since `GMFactorSketch` maintains $2K$ times more hash tables than the CM sketch, we increase the number of bins in the CM sketch to $m' = 2Km$, and get the following upper bound:

$$P_{\text{CM}}(x) \leq \frac{1}{2}\left[\prod_{k=2}^{K+1} C_k\right]\left(1 + \frac{e}{Km}\left[\prod_{k=2}^{K+1} \frac{1}{C_k}\right]\right) . \tag{11}$$

Now both `GMFactorSketch` and the CM sketch consume the same space, and their error bounds are functions of m.

In general, the upper bound in (9) seems to be tighter than that in (11) for the following reason. The bound in (9) involves K potentially large values $1/C_k$, each of which can be offset by a potentially small $1/m$. On the other hand, all values $1/C_k$ in (11) are offset only by single $1/m$.

Before we start our proof, note that both bounds in (9) and (11) involve $\frac{1}{2}\left[\prod_{k=2}^{K+1} C_k\right]$. Therefore, we can divide both bounds by this constant and get that the upper bound in (9) is tighter than that in (11) when:

$$1 + \frac{e}{Km}\left[\prod_{k=2}^{K+1} \frac{1}{C_k}\right] > \prod_{k=2}^{K+1}\left(1 + \frac{2e}{C_k m}\right) \tag{12}$$

Now we rewrite each term $(1 + 2e/(C_k m))$ on the right-hand side as $(1/C_k)(C_k + 2e/m)$ and multiply both sides by $\prod_{k=2}^{K+1} C_k$. Then we omit $\prod_{k=2}^{K+1} C_k$ from the left-hand side and get that event (12) can happen only if:

$$\frac{e}{Km} > \prod_{k=2}^{K+1}\left(C_k + \frac{2e}{m}\right) . \tag{13}$$

If C_k is close to one for all $k \in [K+1] - \{1\}$, the right-hand side of (13) is at least one and we get that m should be smaller than e/K. This result is impractical since K is usually much larger than e and we require that $m \geq 1$. To make progress, we restrict our analysis to a class of x. In particular, let $C_k \leq 1/2$ for all $k \in [K+1] - \{1\}$. Then we can bound the right-hand side of (13) from above as:

$$\prod_{k=2}^{K+1}\left(C_k + \frac{2e}{m}\right) \leq \left(\frac{1}{2}\right)^K \left(1 + \frac{4e}{m}\right)^K \leq e\left(\frac{1}{2}\right)^K$$

when $m \geq 4eK$. The assumption on m is not particularly strong, since Theorem 4 says that we get good multiplicative approximations to $\bar{P}(x)$ only if $m = \Omega(K)$. Now we apply the above upper bound to inequality (13) and rearrange it as $2^K/K > m$. Since $2^K/K$ is exponential in K, we get that the bound in (9) is tighter than that in (11) for a wide range of m and any x where $C_k \leq 1/2$ for all $k \in [K+1] - \{1\}$. Our result is summarized below.

Theorem 5. *Let P be the distribution in (8) and x be such that $P_k(x_k \mid x_1) \leq 1/2$ for all $k \in [K+1] - \{1\}$. Let $m \geq 4eK$ and $m' = 2Km$. Then for any $m < 2^K/K$, the error bound of* `GMFactorSketch` *is tighter than that of the CM sketch at the same space. More precisely:*

$$P(x)\prod_{k=2}^{K+1}(1+\varepsilon_k) \leq P(x) + \frac{e}{m'} ,$$

where ε_k are defined in Theorem 4.

The above result is quite practical. Suppose that $K = 32$. Then our upper bound is tighter for any m such that:

$$4eK < 348 \leq m \leq 2^{27} = 2^{32}/32 = 2^K/K .$$

We also note that Theorem 5 applies in at least $2(M-1)^K$ points x in any naive Bayes model in (8). The fraction of these points can be bounded from below as:

$$\begin{aligned} (M-1)^K/M^K &= \exp[K\log(M-1) - K\log M] \\ &\geq \exp[-K/(M-1)] \\ &\geq 1 - K/(M-1) . \end{aligned}$$

In our motivating problems, $M \approx 10^5$ and $K \approx 100$. In this setting, the error bound of `GMFactorSketch` is guaranteed to be tighter than that of the CM sketch in at least 99.9% of x, for any naive Bayes model in (8).

6. Experiments

In this section, we compare our algorithms (Section 4) and the CM sketch on the synthetic problem in Section 5, and also on a real-world problem in online advertising.

6.1. Synthetic Problem

We consider the naive Bayes model in (8) and set:

$$\begin{aligned} P_1(1) = P_1(2) = 0.5 \\ \forall i \in [N] : P_k(i \mid 1) = 1/N \\ \forall i \in [M] - [N] : P_k(i \mid 1) = 0 \\ \forall i \in [N] : P_k(i \mid 2) = 0 \\ \forall i \in [M] - [N] : P_k(i \mid 2) = 1/(M - N)\,, \end{aligned}$$

where $k \in [K+1] - \{1\}$ and $N \ll M$. The model defines the following distribution over vectors $x = (x_1, \dots, x_K)$: when $x_1 = 1$, $P(x) = 0.5N^{-K}$ and we call these examples *heavy*; and when $x_1 = 2$, $P(x) = 0.5(M - N)^{-K}$ and we call these examples *light*. The heavy examples are much more likely when $N \ll M$. We set $M = 2^{16}$.

All compared algorithms are trained on 1M i.i.d. examples from distribution P and tested on 500k i.i.d. heavy examples from P. We report the fraction of imprecise estimates of P as a function of space and compare the algorithms at the same space. We say that the estimate of $P(x)$ is *precise* when $(1/e)P(x) \leq \hat{P}(x) \leq eP(x)$. When the sample size n is large, both $\bar{P} \to P$ and $\tilde{P} \to P$, and this is a fair way of evaluating both our algorithms and the CM sketch. We choose $d = 5$. We observe similar trends for other values of d and do not report them due to space constraints. All results are averaged over 20 runs.

6.2. Easy Synthetic Problem

We choose $K = 4$ and $N = 8$, and get that $P(x) = 2^{-13}$ for any heavy x. In this problem, the CM sketch can approximate $P(x)$ within a multiplicative factor of e for any heavy x in about 2^{13} space. This space is small and therefore our problem is *easy for the CM sketch* in practice.

Our results are reported in Figure 1a. We observe that all of our algorithms outperform the CM sketch on this problem. In particular, P_{CM} approximates P well for any heavy x in 2^{15} space. Our algorithms achieve the same quality of the approximation in less than 2^{13} space. `GMFactorSketch` consumes 2^{10} space, almost two orders of magnitude improvement over the CM sketch.

6.3. Hard Synthetic Problem

We select $K = 32$ and $N = 64$, and get that $P(x) = 2^{-193}$ for any heavy x. In this problem, the CM sketch can approximate $P(x)$ within a multiplicative factor of e for any heavy x in about 2^{193} space. This space is unrealistically large and therefore our problem is *hard for the CM sketch* in practice.

Our results are reported in Figure 1b and we observe three major trends. First, the CM sketch performs poorly. Second, as in Section 6.3, our algorithms outperform the CM sketch. Finally, when the fraction of imprecise estimates is small, our algorithms perform as suggested by our theory. `GMHash` is inferior to `GMSketch`, which is further inferior to `GMFactorSketch`.

6.4. Real-World Problem

We also evaluate our algorithms on a real-world problem where the goal is to estimate the probability of a page view from high-cardinality data. We take one day of page views of a medium-sized customer of *Adobe Marketing Cloud*[2], which is about 1.5M data points, and extract six variables with each page view: PAGE URL, STARTING PAGE URL, COUNTRY, CITY, CAMPAIGN, and BROWSER. Variable PAGE URL has the highest cardinality and takes on about 30k values. We approximate the probability distribution P over our variables by a naive Bayes model, where the class variable is $X_1 = $ COUNTRY. This approximation is quite reasonable in practice since the location of the user tends to drive his or her behavior.

All compared algorithms are trained on 1M i.i.d. examples from P and tested on its subset of heavy examples. We say that the example x is *heavy* when $P(x) > 10^{-6}$. The rest of the setup is identical to that in Section 6.1.

Our results are reported in Figure 1c. We observe the same trends as in Section 6.3. The CM sketch performs poorly, and our algorithms outperform it at the same space for any space between 2^{12} and 2^{24}. We also note that none of the compared algorithms achieves zero mistakes. This is because our sample size n is not sufficiently large to approximate P well. Therefore, even if $\hat{P} = \bar{P}$, our algorithms would still make a small fraction of mistakes.

7. Conclusions

We propose several algorithms that approximate the MLE of distribution P from a stream of n observations $(x^{(t)})_{t=1}^n$ conditioned on a graphical model $\mathcal{G}$ in the space independent of the cardinality of the variables in $\mathcal{G}$. Our best error bound is provably tighter than that of the CM sketch on a class of naive Bayes models. We evaluate our algorithms on synthetic and real-world problems, and report an order of magnitude improvements in space complexity over the CM sketch at the same quality of the approximation.

The MLE is a fundamental approach to estimating the parameters of graphical models (Jensen, 1996). We propose

[2]http://www.adobe.com/marketing-cloud.html

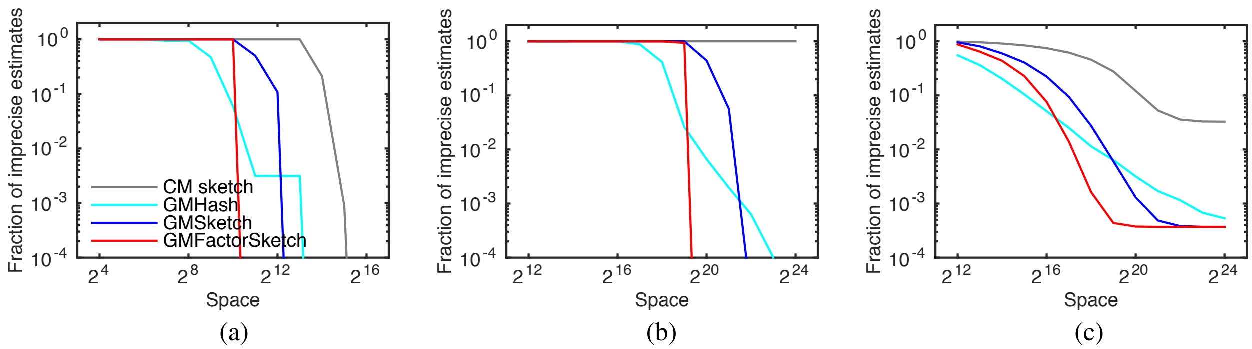

Figure 1. **a**. Evaluation of the CM sketch, `GMHash`, `GMSketch`, and `GMFactorSketch` on the easy problem in Section 6.2. **b**. Evaluation on the hard problem in Section 6.3. **c**. Evaluation on the real-world problem in Section 6.4.

and analyze space-efficient approximations to this estimation procedure with high-cardinality variables, which allow application of graphical models to many large-scale practical problems. Although we focus on the problem of estimating $\bar{P}(x)$, the probability at a single point x, note that our models are combinations of Bayesian networks, and thus can answer any marginal query $P(Y = y)$, where Y is a subset of variables with values y. We do not bound the errors of such queries and leave this for future work.

References

Belle, Vaishak, den Broeck, Guy Van, and Passerini, Andrea. Hashing-based approximate probabilistic inference in hybrid domains. In *Proceedings of the 31th Conference on Uncertainty in Artificial Intelligence*, 2015.

Charikar, Moses, Chen, Kevin, and Farach-Colton, Martin. Finding frequent items in data streams. *Theoretical Computer Science*, 312(1):3–15, 2004.

Cormode, Graham and Muthukrishnan, S. An improved data stream summary: The count-min sketch and its applications. *Journal of Algorithms*, 55(1):58–75, 2005a.

Cormode, Graham and Muthukrishnan, S. What's hot and what's not: Tracking most frequent items dynamically. *ACM Transactions on Database Systems*, 30(1): 249–278, 2005b.

Flajolet, Philippe and Martin, G. Nigel. Probabilistic counting algorithms for data base applications. *Journal of Computer and System Sciences*, 31(2):182–209, 1985.

Flajolet, Philippe, Fusy, Eric, Gandouet, Olivier, and Meunier, Frederic. Hyperloglog: The analysis of a near-optimal cardinality estimation algorithm. In *Proceedings of the 2007 Conference on Analysis of Algorithms*, pp. 127–146, 2007.

Friedman, Nir, Geiger, Dan, and Goldszmidt, Moises. Bayesian network classifiers. *Machine learning*, 29(2-3):131–163, 1997.

Jensen, Finn V. *An introduction to Bayesian networks*, volume 210. UCL press London, 1996.

Koller, Daphne and Friedman, Nir. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Lafferty, John, McCallum, Andrew, and Pereira, Fernando CN. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

Liberty, Edo. Simple and deterministic matrix sketching. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 581–588, 2013.

McDonald, Ryan and Pereira, Fernando. Identifying gene and protein mentions in text using conditional random fields. *BMC bioinformatics*, 6(Suppl 1):S6, 2005.

McGregor, Andrew and Vu, Hoa. Evaluating Bayesian networks via data streams. In *Proceedings of the 21st International Conference on Computing and Combinatorics*, pp. 731–743, 2015.

Misra, Jayadev and Gries, David. Finding repeated elements. *Science of Computer Programming*, 2(2):143–152, 1982.

Murphy, Kevin, Torralba, Antonio, Freeman, William, et al. Using the forest to see the trees: a graphical model relating features, objects and scenes. *Advances in neural information processing systems*, 16:1499–1506, 2003.

Muthukrishnan, S. Data streams: Algorithms and applications. *Foundations and Trends in Theoretical Computer Science*, 2005.

Wainwright, Martin J and Jordan, Michael I. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2): 1–305, 2008.

Woodruff, David. Low rank approximation lower bounds in row-update streams. In *Advances in Neural Information Processing Systems 27*, pp. 1781–1789, 2014.

A. Proofs of Main Theorems

A.1. Proof of Theorem 2

First, we prove a supplementary claim that the number of bins m can be chosen such that:

$$[\bar{P}_1(x_1) - \varepsilon_1] \prod_{k=2}^{K} [\bar{P}_k(x_k \mid x_{\mathsf{pa}(k)}) - \varepsilon_k] \leq \hat{P}(x) \leq [\bar{P}_1(x_1) + \varepsilon_1] \prod_{k=2}^{K} [\bar{P}_k(x_k \mid x_{\mathsf{pa}(k)}) + \varepsilon_k] \tag{14}$$

holds with probability of at least $3/4$ for any $\varepsilon_1, \dots, \varepsilon_K > 0$. Then we choose appropriate $\varepsilon_1, \dots, \varepsilon_K$.

To prove that (14) holds, it is sufficient to show that the following K inequalities:

$$|\hat{P}_1(x_1) - \bar{P}_1(x_1)| \leq \varepsilon_1 \tag{15}$$

$$\forall k \in [K] - \{1\} : |\hat{P}_k(x_k \mid x_{\mathsf{pa}(k)}) - \bar{P}_k(x_k \mid x_{\mathsf{pa}(k)})| \leq \varepsilon_k \tag{16}$$

hold jointly with probability of at least $3/4$. Now we bound the probability that either of the inequalities do not hold.

Clearly $\hat{P}_1(x_1) - \bar{P}_1(x_1) \geq 0$. Therefore, the probability that (15) does not hold is bounded by Lemma 2 as:

$$P(|\hat{P}_1(x_1) - \bar{P}_1(x_1)| > \varepsilon_1) = P(\hat{P}_1(x_1) - \bar{P}_1(x_1) > \varepsilon_1) < \frac{1}{m\varepsilon_1}\,. \tag{17}$$

Now we fix $k \in [K] - \{1\}$ and bound the probability that (16) does not hold:

$$P(|\hat{P}_k(x_k \mid x_{\mathsf{pa}(k)}) - \bar{P}_k(x_k \mid x_{\mathsf{pa}(k)})| > \varepsilon_k) =$$
$$P\left(\left|\frac{\bar{c}_k(h_k(x_k + M(x_{\mathsf{pa}(k)} - 1)))}{c_{\mathsf{pa}(k)}(h_{\mathsf{pa}(k)}(x_{\mathsf{pa}(k)}))} - \frac{\sum_{t=1}^{n} \mathbb{1}\left\{x_k^{(t)} = x_k, x_{\mathsf{pa}(k)}^{(t)} = x_{\mathsf{pa}(k)}\right\}}{\sum_{t=1}^{n} \mathbb{1}\left\{x_{\mathsf{pa}(k)}^{(t)} = x_{\mathsf{pa}(k)}\right\}}\right| > \varepsilon_k\right)\,.$$

By Lemma 1, the necessary conditions for $|\hat{P}_k(x_k \mid x_{\mathsf{pa}(k)}) - \bar{P}_k(x_k \mid x_{\mathsf{pa}(k)})| > \varepsilon_k$ are:

$$\frac{1}{n} c_{\mathsf{pa}(k)}(h_{\mathsf{pa}(k)}(x_{\mathsf{pa}(k)})) - \frac{1}{n}\sum_{t=1}^{n} \mathbb{1}\left\{x_{\mathsf{pa}(k)}^{(t)} = x_{\mathsf{pa}(k)}\right\} > \varepsilon_k \alpha_k$$
$$\frac{1}{n} \bar{c}_k(h_k(x_k + M(x_{\mathsf{pa}(k)} - 1))) - \frac{1}{n}\sum_{t=1}^{n} \mathbb{1}\left\{x_k^{(t)} = x_k, x_{\mathsf{pa}(k)}^{(t)} = x_{\mathsf{pa}(k)}\right\} > \varepsilon_k \alpha_k\,,$$

where $\alpha_k = \bar{P}_{\mathsf{pa}(k)}(x_{\mathsf{pa}(k)})$. The first event happens when the denominator of $\hat{P}_k(x_k \mid x_{\mathsf{pa}(k)})$ increases significantly when compared to that of $\bar{P}_k(x_k \mid x_{\mathsf{pa}(k)})$. The second event happens when the numerator increases significantly.

Now we show that the above events are unlikely. The probability of the first event is bounded by Lemma 2 as:

$$P\left(\frac{1}{n} c_{\mathsf{pa}(k)}(h_{\mathsf{pa}(k)}(x_{\mathsf{pa}(k)})) - \frac{1}{n}\sum_{t=1}^{n} \mathbb{1}\left\{x_{\mathsf{pa}(k)}^{(t)} = x_{\mathsf{pa}(k)}\right\} > \varepsilon_k \alpha_k\right) < \frac{1}{m\varepsilon_k\alpha_k} \tag{18}$$

for $X = X_{\mathsf{pa}(k)}$, $h = h_{\mathsf{pa}(k)}$, and $\varepsilon = \varepsilon_k \alpha_k$. The probability of the second event is bounded by Lemma 2 as:

$$P\left(\frac{1}{n} \bar{c}_k(h_k(x_k + M(x_{\mathsf{pa}(k)} - 1))) - \frac{1}{n}\sum_{t=1}^{n} \mathbb{1}\left\{x_k^{(t)} = x_k, x_{\mathsf{pa}(k)}^{(t)} = x_{\mathsf{pa}(k)}\right\} > \varepsilon_k \alpha_k\right) < \frac{1}{m\varepsilon_k\alpha_k} \tag{19}$$

for $X = X_k + M(X_{\mathsf{pa}(k)} - 1)$, $h = h_k$, and $\varepsilon = \varepsilon_k \alpha_k$. Now we chain inequalities (17), (18), and (19); and get by the union that inequalities (15) and (16) do not hold with probability of at most:

$$\frac{1}{m\varepsilon_1} + \sum_{k=1}^{K} \frac{2}{m\varepsilon_k\alpha_k} < \frac{1}{m}\sum_{k=1}^{K} \frac{2}{\varepsilon_k\alpha_k}\,,$$

where $\alpha_1 = 1$. This probability is at most $1/4$ for $m \geq 4 \sum_{k=1}^{K} \frac{2}{\varepsilon_k \alpha_k}$. This concludes the proof of (14).

Now we choose appropriate $\varepsilon_1, \dots, \varepsilon_K$. In particular, let $\varepsilon_k = 8K/(\alpha_k m)$ for any $k \in [K]$. Note that this setting is valid for any $m \geq 1$ since:

$$m \geq 4 \sum_{k=1}^{K} \frac{2}{\varepsilon_k \alpha_k} = 4 \sum_{k=1}^{K} \frac{m}{4K} = m\,.$$

Under this assumption, the upper bound in (14) can be written as:

$$\begin{aligned}
\hat{P}(x) &\leq [\bar{P}_1(x_1) + \varepsilon_1] \prod_{k=2}^{K} [\bar{P}_k(x_k \mid x_{\mathsf{pa}(k)}) + \varepsilon_k] \\
&= \left[\bar{P}_1(x_1) + \frac{8K}{\alpha_1 m}\right] \prod_{k=2}^{K} \left[\bar{P}_k(x_k \mid x_{\mathsf{pa}(k)}) + \frac{8K}{\alpha_k m}\right] \\
&= \left[\bar{P}_1(x_1) \prod_{k=2}^{K} \bar{P}_k(x_k \mid x_{\mathsf{pa}(k)})\right] \left[1 + \frac{8K}{\bar{P}_1(x_1) m}\right] \prod_{k=2}^{K} \left[1 + \frac{8K}{\bar{P}_k(x_k, x_{\mathsf{pa}(k)}) m}\right].
\end{aligned}$$

Along the same lines, the lower bound in (14) can be written as:

$$\begin{aligned}
\hat{P}(x) &\geq [\bar{P}_1(x_1) - \varepsilon_1] \prod_{k=2}^{K} [\bar{P}_k(x_k \mid x_{\mathsf{pa}(k)}) - \varepsilon_k] \\
&= \left[\bar{P}_1(x_1) - \frac{8K}{\alpha_1 m}\right] \prod_{k=2}^{K} \left[\bar{P}_k(x_k \mid x_{\mathsf{pa}(k)}) - \frac{8K}{\alpha_k m}\right] \\
&= \left[\bar{P}_1(x_1) \prod_{k=2}^{K} \bar{P}_k(x_k \mid x_{\mathsf{pa}(k)})\right] \left[1 - \frac{8K}{\bar{P}_1(x_1) m}\right] \prod_{k=2}^{K} \left[1 - \frac{8K}{\bar{P}_k(x_k, x_{\mathsf{pa}(k)}) m}\right].
\end{aligned}$$

This concludes our proof.

A.2. Proof of Theorem 3

Algorithm `GMSketch` estimates the probability as a median of d probabilities:

$$\hat{P}(x) = \underset{i \in [d]}{\text{median}}\ \hat{P}^i(x)\,,$$

each of which is estimated by a random instance of `GMHash`. We bound the probability that $\hat{P}(x)$ is a good approximation of $\bar{P}(x)$:

$$\bar{P}(x) \prod_{k=1}^{K} (1 - \varepsilon_k) \leq \hat{P}(x) \leq \bar{P}(x) \prod_{k=1}^{K} (1 + \varepsilon_k)\,,$$

where ε_k are defined in Theorem 2, using the so-called median trick. Let:

$$Z_i = \mathbb{1}\left\{\bar{P}(x) \prod_{k=1}^{K} (1 - \varepsilon_k) \leq \hat{P}^i(x) \leq \bar{P}(x) \prod_{k=1}^{K} (1 + \varepsilon_k)\right\}$$

indicate the event that $\hat{P}^i(x)$ approximates $\bar{P}(x)$ well. Let $\bar{Z} = \frac{1}{d}\sum_{i=1}^{d} Z_i$ and $\mathbb{E}\left[\bar{Z}\right] \geq 1/2$, where the expectation is with respect to the random hashes $h^1, \dots, h^d$. Then by Hoeffding's inequality:

$$P(\mathbb{E}\left[\bar{Z}\right] - \bar{Z} > \mathbb{E}\left[\bar{Z}\right] - 1/2) < \exp[-2(\mathbb{E}\left[\bar{Z}\right] - 1/2)^2 d]\,,$$

where $\mathbb{E}\left[\bar{Z}\right] - \bar{Z} > \mathbb{E}\left[\bar{Z}\right] - 1/2$ is the event that $\hat{P}(x)$ does not approximate $\bar{P}(x)$ well. By Theorem 2, we know that $\mathbb{E}\left[\bar{Z}\right] \geq 3/4$ and therefore:

$$P(\mathbb{E}\left[\bar{Z}\right] - \bar{Z} > \mathbb{E}\left[\bar{Z}\right] - 1/2) < \exp[-2(3/4 - 1/2)^2 d] = \exp[-d/8]\,.$$

Now we choose $d \geq 8\log(1/\delta)$ and get that $\hat{P}(x)$ is a not a good approximation of $\bar{P}(x)$ with probability of at most δ.

A.3. Proof of Theorem 4

The key idea of this proof is similar to that of Theorem 2. First, we prove a supplementary claim that the number of bins m can be chosen such that:

$$[\bar{P}_1(x_1) - \varepsilon_1] \prod_{k=2}^{K} [\bar{P}_k(x_k \mid x_{\mathsf{pa}(k)}) - \varepsilon_k] \leq \hat{P}(x) \leq [\bar{P}_1(x_1) + \varepsilon_1] \prod_{k=2}^{K} [\bar{P}_k(x_k \mid x_{\mathsf{pa}(k)}) + \varepsilon_k] \tag{20}$$

holds with probability of at least $1 - \delta$ for any $\varepsilon_1, \dots, \varepsilon_K > 0$. Then we choose appropriate $\varepsilon_1, \dots, \varepsilon_K$.

To prove that (20) holds, it is sufficient to show that the following K inequalities:

$$\begin{aligned} |\hat{P}_1(x_1) - \bar{P}_1(x_1)| &\leq \varepsilon_1 \\ \forall k \in [K] - \{1\} : |\hat{P}_k(x_k \mid x_{\mathsf{pa}(k)}) - \bar{P}_k(x_k \mid x_{\mathsf{pa}(k)})| &\leq \varepsilon_k \end{aligned}$$

hold jointly with probability of at least $1 - \delta$. By Lemma 1 and the union bound, this is equivalent to showing that each of the following inequalities:

$$\begin{aligned} \hat{P}_1(x_1) - \bar{P}_1(x_1) &\leq \varepsilon_1 \alpha_1 \\ \forall k \in [K] - \{1\} : \hat{P}_{\mathsf{pa}(k)}(x_{\mathsf{pa}(k)}) - \bar{P}_{\mathsf{pa}(k)}(x_{\mathsf{pa}(k)}) &\leq \varepsilon_k \alpha_k \\ \forall k \in [K] - \{1\} : \hat{P}_k(x_k, x_{\mathsf{pa}(k)}) - \bar{P}_k(x_k, x_{\mathsf{pa}(k)}) &\leq \varepsilon_k \alpha_k \end{aligned}$$

is violated with probability of at most $\delta/(2K)$, where $\alpha_1 = 1$ and $\alpha_k = \bar{P}_{\mathsf{pa}(k)}(x_{\mathsf{pa}(k)})$ for any $k \in [K] - \{1\}$. Now note that each $\hat{P}$ is the CM sketch of the corresponding $\bar{P}$. So, by Theorem 1 of Cormode & Muthukrishnan (2005a), each of the above inequalities is violated with probability of at most $\delta/(2K)$ when the number of hash functions $d \geq \log(2K/\delta)$ and the number of hashing bins m satisfies:

$$\begin{aligned} m &\geq \frac{e}{\varepsilon_1 \alpha_1} \\ \forall k \in [K] - \{1\} : m &\geq \frac{e}{\varepsilon_k \alpha_k}\,. \end{aligned}$$

To satisfy the above inequalities, we select appropriate $\varepsilon_1, \dots, \varepsilon_K$. In particular, let $\varepsilon_k = e/(\alpha_k m)$ for any $k \in [K]$. This setting is valid for any $m \geq 1$ and $k \in [K]$ since:

$$m \geq \frac{e}{\varepsilon_k \alpha_k} = m\,.$$

Under this assumption, the upper bound in (20) can be written as:

$$\begin{aligned} \hat{P}(x) &\leq [\bar{P}_1(x_1) + \varepsilon_1] \prod_{k=2}^{K} [\bar{P}_k(x_k \mid x_{\mathsf{pa}(k)}) + \varepsilon_k] \\ &= \left[\bar{P}_1(x_1) + \frac{e}{\alpha_1 m}\right] \prod_{k=2}^{K} \left[\bar{P}_k(x_k \mid x_{\mathsf{pa}(k)}) + \frac{e}{\alpha_k m}\right] \\ &= \left[\bar{P}_1(x_1) \prod_{k=2}^{K} \bar{P}_k(x_k \mid x_{\mathsf{pa}(k)})\right] \left[1 + \frac{e}{\bar{P}_1(x_1) m}\right] \prod_{k=2}^{K} \left[1 + \frac{e}{\bar{P}_k(x_k, x_{\mathsf{pa}(k)}) m}\right]\,. \end{aligned}$$

Along the same lines, the lower bound in (20) can be written as:

$$\begin{aligned}\hat{P}(x) &\geq [\bar{P}_1(x_1) - \varepsilon_1] \prod_{k=2}^{K} [\bar{P}_k(x_k \mid x_{\mathsf{pa}(k)}) - \varepsilon_k] \\ &= \left[\bar{P}_1(x_1) - \frac{e}{\alpha_1 m}\right] \prod_{k=2}^{K} \left[\bar{P}_k(x_k \mid x_{\mathsf{pa}(k)}) - \frac{e}{\alpha_k m}\right] \\ &= \left[\bar{P}_1(x_1) \prod_{k=2}^{K} \bar{P}_k(x_k \mid x_{\mathsf{pa}(k)})\right] \left[1 - \frac{e}{\bar{P}_1(x_1) m}\right] \prod_{k=2}^{K} \left[1 - \frac{e}{\bar{P}_k(x_k, x_{\mathsf{pa}(k)}) m}\right] .\end{aligned}$$

This concludes our proof.

B. Technical Lemmas

Lemma 1. *Let:*

$$\left| \frac{u_h}{v_h} - \frac{u}{v} \right| > \varepsilon$$

for any $u_h \geq u$, $v_h \geq v$, $v \geq u$, and $v \geq \alpha n$. Then either $v_h - v > \varepsilon \alpha n$ or $u_h - u > \varepsilon \alpha n$.

Proof. The proof is by contradiction. First, we note that $\left| \frac{u_h}{v_h} - \frac{u}{v} \right| > \varepsilon$ implies that either:

$$\frac{u_h}{v_h} - \frac{u}{v} > \varepsilon \qquad \text{or} \qquad \frac{u}{v} - \frac{u_h}{v_h} > \varepsilon .$$

We argue that $u_h/v_h - u/v > \varepsilon$ implies $u_h - u > \varepsilon \alpha n$. Suppose that this was not true. Then the opposite must be true, $u_h/v_h - u/v > \varepsilon$ and $u_h - u \leq \varepsilon \alpha n$. We derive contradiction by bounding ε from above as:

$$\varepsilon < \frac{u_h}{v_h} - \frac{u}{v} = \underbrace{\frac{v}{v_h}}_{\leq 1} \frac{u_h}{v} - \frac{u}{v} \leq \frac{u_h - u}{v} \leq \frac{u_h - u}{\alpha n} .$$

Now we argue that $u/v - u_h/v_h > \varepsilon$ implies $v_h - v > \varepsilon \alpha n$. Suppose that this was not true. Then the opposite must be true, $u/v - u_h/v_h > \varepsilon$ and $v_h - v \leq \varepsilon \alpha n$. We derive contradiction by bounding ε from above as:

$$\varepsilon < \frac{u}{v} - \frac{u_h}{v_h} = \frac{u}{v} - \underbrace{\frac{u_h}{u}}_{\geq 1} \frac{u}{v_h} \leq \underbrace{\frac{u}{v}}_{\leq 1} \frac{v_h - v}{v_h} \leq \frac{v_h - v}{\alpha n} .$$

The last step follows from $v_h \geq v \geq \alpha n$. This concludes our proof. ■

Lemma 2. *Let X be a discrete random variable on $\mathbb{N}$ and $(x^{(t)})_{t=1}^n$ be its n observations. Let $h : \mathbb{N} \to [m]$ be any random hash function. Then for any $x \in \mathbb{N}$, $m \geq 1$, and $\varepsilon \in (0, 1)$:*

$$P\left(\frac{1}{n} \sum_{t=1}^{n} \mathbb{1}\left\{h(x^{(t)}) = h(x)\right\} - \frac{1}{n} \sum_{t=1}^{n} \mathbb{1}\left\{x^{(t)} = x\right\} > \varepsilon\right) < \frac{1}{m\varepsilon} ,$$

where the randomness is with respect to h.

Proof. Clearly:

$$\frac{1}{n} \sum_{t=1}^{n} \mathbb{1}\left\{h(x^{(t)}) = h(x)\right\} - \frac{1}{n} \sum_{t=1}^{n} \mathbb{1}\left\{x^{(t)} = x\right\} \geq 0$$

because $x^{(t)} = x$ implies that $h(x^{(t)}) = h(x)$ for any $h : \mathbb{N} \to [m]$. Therefore, we can apply Markov's inequality and get:

$$P\left(\frac{1}{n}\sum_{t=1}^{n}\mathbb{1}\left\{h(x^{(t)}) = h(x)\right\} - \frac{1}{n}\sum_{t=1}^{n}\mathbb{1}\left\{x^{(t)} = x\right\} > \varepsilon\right) < \frac{1}{\varepsilon n}\mathbb{E}\left[\sum_{t=1}^{n}\mathbb{1}\left\{h(x^{(t)}) = h(x)\right\} - \sum_{t=1}^{n}\mathbb{1}\left\{x^{(t)} = x\right\}\right]$$
$$= \frac{1}{\varepsilon n}\sum_{t=1}^{n}\mathbb{E}\left[\mathbb{1}\left\{h(x^{(t)}) = h(x),\ x^{(t)} \neq x\right\}\right].$$

where the last equality is by the linearity of expectation. Because h is random, the probability that $h(x^{(t)}) = h(x)$ when $x^{(t)} \neq x$ is $1/m$. Therefore, $\mathbb{E}\left[\mathbb{1}\{h(x^{(t)}) = h(x),\ x^{(t)} \neq x\}\right] \leq 1/m$ and we conclude that:

$$P\left(\frac{1}{n}\sum_{t=1}^{n}\mathbb{1}\left\{h(x^{(t)}) = h(x)\right\} - \frac{1}{n}\sum_{t=1}^{n}\mathbb{1}\left\{x^{(t)} = x\right\} > \varepsilon\right) < \frac{1}{\varepsilon m}.$$

■