

Graphical Model Sketch

Branislav Kveton¹(✉), Hung Bui¹, Mohammad Ghavamzadeh¹,
Georgios Theodorou¹, S. Muthukrishnan², and Siqi Sun³

¹ Adobe Research, San Jose, CA, USA

{kveton,hubui,ghavamza,theochar}@adobe.com

² Department of Computer Science, Rutgers, New Brunswick, NJ, USA

muthu@cs.rutgers.edu

³ TTI, Chicago, IL, USA

siqi.sun@ttic.edu

Abstract. Structured high-cardinality data arises in many domains, and poses a major challenge for both modeling and inference. Graphical models are a popular approach to modeling structured data but they are unsuitable for high-cardinality variables. The count-min (CM) sketch is a popular approach to estimating probabilities in high-cardinality data but it does not scale well beyond a few variables. In this work, we bring together the ideas of graphical models and count sketches; and propose and analyze several approaches to estimating probabilities in structured high-cardinality streams of data. The key idea of our approximations is to use the structure of a graphical model and approximately estimate its factors by “sketches”, which hash high-cardinality variables using random projections. Our approximations are computationally efficient and their space complexity is independent of the cardinality of variables. Our error bounds are multiplicative and significantly improve upon those of the CM sketch, a state-of-the-art approach to estimating probabilities in streams. We evaluate our approximations 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 numerous domains, and poses a major challenge for modeling and inference. A common goal in online advertising is to estimate the probability of events, such as page views, over multiple high-cardinality variables, such as the location of the user, the referring page, and the purchased product. A common goal in natural language processing is to estimate the probability of n -grams over a dictionary of 100k words. Graphical models [9] are a popular approach to modeling multivariate data. However, when the cardinality of random variables is high, they are expensive to store and reason with. For instance, a graphical model over two variables with $M = 10^5$ values each may consume $M^2 = 10^{10}$ space.

A *sketch* [17] is a data structure that summarizes streams of data such that any two sketches of individual streams can be combined space efficiently into

the sketch of the combined stream. Numerous problems can be solved efficiently by surprisingly simple sketches, such as estimating the frequency of values in streams [3, 4, 15], finding heavy hitters [5], estimating the number of unique values [7, 8], or even approximating low-rank matrices [12, 18]. In this work, we sketch a graphical model in a small space. Let $(x^{(t)})_{t=1}^n$ be a stream of n observations from some distribution P , where $x^{(t)} \in [M]^K$ is a K -dimensional vector and P factors 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 \mathcal{G} . Then our goal is to approximate \bar{P} with \hat{P} such that $\hat{P}(x) \approx \bar{P}(x)$ for any $x \in [M]^K$ with at least $1 - \delta$ probability; in the space that does not depend on the cardinality M of the variables in \mathcal{G} . In our motivating examples, x is an n -gram or the feature vector associated with page views.

This paper makes three contributions. First, we propose and carefully analyze three natural approximations to the MLE in graphical models with high-cardinality variables. The key idea of our approximations is to leverage the structure of the graphical model \mathcal{G} and approximately estimate its factors by “sketches”. Therefore, we refer to our approximations as *graphical model sketches*. Our best approximation, **GMFactorSketch**, guarantees that $\hat{P}(x)$ is a constant-factor multiplicative approximation to $\bar{P}(x)$ for any x with probability of at least $1 - \delta$ in $O(K^2 \log(K/\delta) \Delta^{-1}(x))$ space, where K is the number of variables and $\Delta(x)$ measures the hardness of query x . The dependence on $\Delta(x)$ is generally unavoidable and we show this in Sect. 5.4. Second, we prove that **GMFactorSketch** yields better approximations than the count-min (CM) sketch [4], a state-of-the-art approach to estimating the frequency of values in streams (Sect. 6). Third, we evaluate our approximations on both synthetic and real-world problems. Our results show that **GMFactorSketch** outperforms the CM sketch and our other approximations, as measured by the error in estimating \bar{P} at the same space.

Our work is related to Matushevych *et al.* [13], who proposed several extensions of the CM sketch, one of which is **GMFactorSketch**. This approximation is not analyzed and it is evaluated only on a graphical model with three variables. We present the first analysis of **GMFactorSketch**, and prove that it is superior to other natural approximations and the CM sketch. We also evaluate **GMFactorSketch** on an order of magnitude larger problems than Matushevych *et al.* [13]. McGregor and Vu [14] proposed and analyzed a space-efficient streaming algorithm that tests if the stream of data is consistent with a graphical model. Several recent papers applied hashing to speeding up inference in graphical models [1, 6]. These papers do not focus on high-cardinality variables and are only loosely related to our work, because of using hashing in graphical models. We also note that the problem of representing conditional probabilities in graphical models efficiently has been studied extensively, as early as in Boutilier *et al.* [2]. Our paper is different from this line of work because we do not assume any sparsity or symmetry in data; and our approximations are suitable for the streaming setting.

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 reviews the two main components of our solutions.

2.1 Count-Min Sketch

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. Suppose that we want to estimate:

$$\tilde{P}(x) = \frac{1}{n} \sum_{t=1}^n \mathbb{1}\{x = x^{(t)}\}, \quad (1)$$

the frequency of observing any x in $(x^{(t)})_{t=1}^n$. This problem can be solved in $O(M^K)$ space, by counting all unique values in $(x^{(t)})_{t=1}^n$. This solution is impractical when K and M are large. Cormode and Muthukrishnan [4] proposed an approximate solution to this problem, the *count-min (CM) sketch*, which estimates $\tilde{P}(x)$ in the space independent of M^K . The sketch consists of d hash tables with m bins, $c \in \mathbb{N}^{d \times m}$. The hash tables are initialized with zeros. At time t , they are updated with observation $x^{(t)}$ as:

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

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

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

Cormode and Muthukrishnan [4] showed 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 $O((1/\varepsilon) \log(1/\delta))$ space. Note that 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 at least $1 - \delta$ probability. The space complexity of P_{CM} is $(e/\varepsilon) \log(1/\delta)$.*

The CM sketch is popular because high-quality approximations, with at most ε error, can be computed in $O(1/\varepsilon)$ space.¹ Other similar sketches, such as Charikar *et al.* [3], require $O(1/\varepsilon^2)$ space.

¹ <https://sites.google.com/site/countminsketch/>.

2.2 Bayesian Networks

Graphical models are a popular tool for modeling and reasoning with random variables [10], and have many applications in computer vision [16] and natural language processing [11]. In this work, we focus on Bayesian networks [9], which are directed graphical models.

A *Bayesian network* is a probabilistic graphical model that represents conditional independencies of random variables by a directed graph. In this work, we define it as a pair (\mathcal{G}, θ) , where \mathcal{G} is a directed graph and θ are its parameters. The graph $\mathcal{G} = (V, E)$ is defined by its nodes $V = \{X_1, \dots, X_K\}$, one for each random variable, and edges E . For simplicity of exposition, we assume that \mathcal{G} is a *tree* and X_1 is its root. We relax this assumption in Sect. 3. Under this assumption, each node X_k for $k \geq 2$ has one parent and the probability of $x = (x_1, \dots, x_K)$ factors as:

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

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

$$P_k(i) = P(X_k = i), \quad P_k(i, j) = P(X_k = i, X_{\text{pa}(k)} = j), \quad P_k(i \mid j) = \frac{P_k(i, j)}{P_{\text{pa}(k)}(j)}.$$

Let $\text{dom}(X_k) = M$ for all $k \in [K]$. Then our graphical model 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:

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

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

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

3 Model

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. Our objective is to approximate $\bar{P}(x)$ in (3), the

frequency of observing x as given by the MLE of P from $(x^{(t)})_{t=1}^n$ conditioned on graphical model \mathcal{G} . This objective naturally generalizes that 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 (Sect. 2.2). Under this assumption, \bar{P} can be represented exactly in $O(KM^2)$ space. This is not feasible in our problems of interest, where typically $M \geq 10^4$.

The key idea in our solutions is to estimate a surrogate parameter $\hat{\theta}$. We estimate $\hat{\theta}$ on the same graphical model as θ . The difference is that $\hat{\theta}$ parameterizes a graphical model where each factor is represented by $O(m)$ hashing bins, where $m \ll M^2$. Our proposed models consume $O(Km)$ space, a significant reduction from $O(KM^2)$; and guarantee that $\hat{P}(x) \approx \bar{P}(x)$ for any $x \in [M]^K$ and observations $(x^{(t)})_{t=1}^n$ up to time n , where we abbreviate $P(X = x \mid \theta, \mathcal{G})$ as $\bar{P}(x)$. More precisely:

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

for any $x \in [M]^K$ with at least $1 - \delta$ probability, where \hat{P} is factored in the same way as \bar{P} . Each term ε_k is $O(1/m)$, where m is the number of hashing bins. Therefore, the quality of our approximations improves as m increases. More precisely, if m is chosen such that $\varepsilon_k \leq 1/K$ for all $k \in [K]$, we get:

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

for $K \geq 2$ since $\prod_{k=1}^K (1 + \varepsilon_k) \leq (1 + 1/K)^K \leq e$ for $K \geq 1$ and $\prod_{k=1}^K (1 - \varepsilon_k) \geq (1 - 1/K)^K \geq 2/(3e)$ for $K \geq 2$. Therefore, $\hat{P}(x)$ is a constant-factor multiplicative approximation to $\bar{P}(x)$. As in the CM sketch, we do not require that $\hat{P}(x)$ sum up to 1.

4 Summary of Main Results

The main contribution of our work is that we propose and analyze three approaches to the MLE in graphical models with high-cardinality variables. Our first proposed algorithm, **GMHash** (Sect. 5.1), approximates $\bar{P}(x)$ as the product of $K - 1$ conditionals and a prior, one for each variable in \mathcal{G} . Each conditional is estimated as a ratio of two hashing bins. **GMHash** guarantees (5) for any $x \in [M]^K$ with at least $1 - \delta$ probability in $O(K^3 \delta^{-1} \Delta^{-1}(x))$ space, where $\Delta(x)$ is a query-specific constant and the number of hashing bins is set as $m = \Omega(K^2 \delta^{-1})$. We discuss $\Delta(x)$ at the end of this section. Since δ is typically small, the dependence on $1/\delta$ is undesirable.

Our second algorithm, **GMSketch** (Sect. 5.2), approximates $\bar{P}(x)$ as the median of d probabilities, each of which is estimated by **GMHash**. **GMSketch** guarantees (5) for any $x \in [M]^K$ with at least $1 - \delta$ probability in $O(K^3 \log(1/\delta) \Delta^{-1}(x))$ space, when we set $m = \Omega(K^2 \Delta^{-1}(x))$.

and $d = \Omega(\log(1/\delta))$. The main advantage over **GMHash** is that the space is $O(\log(1/\delta))$ instead of $O(1/\delta)$.

Our last algorithm, **GMFactorSketch** (Sect. 5.3), approximates $\bar{P}(x)$ as the product of $K-1$ conditionals and a prior, one for each variable. Each conditional is estimated as a ratio of two count-min sketches. **GMFactorSketch** guarantees (5) for any $x \in [M]^K$ with at least $1-\delta$ probability in $O(K^2 \log(K/\delta) \Delta^{-1}(x))$ space, when we set $m = \Omega(K \Delta^{-1}(x))$ and $d = \Omega(\log(K/\delta))$. The key improvement over **GMSketch** is that the space is $O(K^2)$ instead of being $O(K^3)$. In summary, **GMFactorSketch** is the best of our proposed solutions. We demonstrate this empirically in Sect. 7.

The query-specific constant $\Delta(x) = \min_{k \in [K] - \{1\}} \bar{P}_k(x_k, x_{\text{pa}(k)})$ is the minimum probability that the values of any variable-parent pair in x co-occur in $(x^{(t)})_{t=1}^n$. This probability can be small and our algorithms are unsuitable for estimating $\bar{P}(x)$ in such cases. Note that this does not imply that $\bar{P}(x)$ cannot be small. Unfortunately, the dependence on $\Delta(x)$ is generally unavoidable and we show this in Sect. 5.4.

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

5 Algorithms and Analysis

All of our algorithms hash the values of each variable in graphical model \mathcal{G} , and each variable-parent pair, to m bins up to d times. We denote the i -th hash function of variable X_k by h_k^i and the associated hash table by $c_k(i, \cdot)$. This hash table approximates $n\bar{P}_k(\cdot)$. The i -th hash function of the variable-parent pair $(X_k, X_{\text{pa}(k)})$ is also h_k^i , and the associated hash table is $\bar{c}_k(i, \cdot)$. This hash table approximates $n\bar{P}_k(\cdot, \cdot)$. Our algorithms differ in how the hash tables are aggregated.

We define the notion of a *hash*, which is a tuple $h = (h_1, \dots, h_K)$ of K randomly drawn hash functions $h_k : \mathbb{N} \rightarrow [m]$, one for each variable in \mathcal{G} . We make the assumption that hashes are pairwise-independent. We say that hashes h^i and h^j are *pairwise-independent* when h_k^i and h_k^j are pairwise-independent for all $k \in [K]$. These kinds of hash functions can be computed fast and stored in a very small space [4].

5.1 Algorithm GMHash

The pseudocode of our first algorithm, **GMHash**, is 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:

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

Algorithm 1. GMHash: Hashed conditionals and priors.**Input:** Point query $x = (x_1, \dots, x_K)$

$$\begin{aligned}
\hat{P}_1(x_1) &\leftarrow \frac{c_1(h_1(x_1))}{n} \\
\text{for all } k = 2, \dots, K \text{ do} \\
\hat{P}_k(x_k \mid x_{\text{pa}(k)}) &\leftarrow \frac{\bar{c}_k(h_k(x_k + M(x_{\text{pa}(k)} - 1)))}{c_{\text{pa}(k)}(h_{\text{pa}(k)}(x_{\text{pa}(k)}))} \\
\hat{P}(x) &\leftarrow \hat{P}_1(x_1) \prod_{k=2}^K \hat{P}_k(x_k \mid x_{\text{pa}(k)})
\end{aligned}$$

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

where $\bar{c}_k(h_k(x_k + M(x_{\text{pa}(k)} - 1)))$ is the number of times that hash function h_k maps $(x_k^{(t)}, x_{\text{pa}(k)}^{(t)})$ to the same bin as $(x_k, x_{\text{pa}(k)})$ in n steps, and $c_k(h_k(x_k))$ is the number of times that h_k maps $x_k^{(t)}$ to the same bin as x_k in n steps. Note that $(x_k, x_{\text{pa}(k)})$ can be represented equivalently as $x_k + M(x_{\text{pa}(k)} - 1)$. The prior $\bar{P}_1(x_1)$ is estimated 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 $k \in [K], y \in [m]$:

$$\begin{aligned}
c_k(y) &\leftarrow c_k(y) + \mathbb{1}\{y = h_k(x_k^{(t)})\}, \\
\bar{c}_k(y) &\leftarrow \bar{c}_k(y) + \mathbb{1}\{y = h_k(x_k^{(t)} + M(x_{\text{pa}(k)}^{(t)} - 1))\}.
\end{aligned}$$

This update takes $O(K)$ time.

GMHash maintains $2K - 1$ hash tables with m bins each, one for each variable and one for each variable-parent pair in \mathcal{G} . Therefore, it consumes $O(Km)$ space. Now we show that \hat{P} is a good approximation of \bar{P} .

Theorem 2. Let \hat{P} be the estimator from Algorithm 1. Let h be a random hash and m be the number of bins in each hash function. 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 at least $1 - \delta$ probability, where:

$$\varepsilon_1 = 2K[\bar{P}_1(x_1)\delta m]^{-1}, \quad \forall k \in [K] - \{1\} : \varepsilon_k = 2K[\bar{P}_k(x_k, x_{\text{pa}(k)})\delta m]^{-1}.$$

Proof. The proof is in Appendix. The key idea is to show that the number of bins m can be chosen such that:

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

Algorithm 2. GMSketch: Median of d GMHash estimates.**Input:** Point query $x = (x_1, \dots, x_K)$ **for all** $i = 1, \dots, d$ **do**

$$\hat{P}_1^i(x_1) \leftarrow \frac{c_1(i, h_1^i(x_1))}{n}$$

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

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

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

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

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

is not likely for any $k \in [K] - \{1\}$ and $\varepsilon_1, \dots, \varepsilon_K > 0$. In other words, we argue that our estimate of each conditional $\bar{P}_k(x_k \mid x_{\text{pa}(k)})$ can be arbitrary precise. By Lemma 1 in Appendix, the necessary conditions for event (6) are:

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

where $\alpha_k = \bar{P}_{\text{pa}(k)}(x_{\text{pa}(k)})$ is the frequency that $X_{\text{pa}(k)} = x_{\text{pa}(k)}$ in $(x^{(t)})_{t=1}^n$. In short, event (6) can happen only if GMHash significantly overestimates either $\bar{P}_{\text{pa}(k)}(x_{\text{pa}(k)})$ or $\bar{P}_k(x_k, x_{\text{pa}(k)})$. We bound the probability of these events using Markov's inequality (Lemma 2 in Appendix) and then get that none of the events in (6) happen with at least $1 - \delta$ probability when the number of hashing bins $m \geq \sum_{k=1}^K (2/(\varepsilon_k \alpha_k \delta))$. Finally, we choose appropriate $\varepsilon_1, \dots, \varepsilon_K$.

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 $O(1/m)$. The accuracy of the approximation depends on the frequency of interaction between the values in x . In particular, if $\bar{P}_k(x_k, x_{\text{pa}(k)})$ is sufficiently large for all $k \in [K] - \{1\}$, the approximation is good even for small m . More precisely, under the assumptions that:

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

all $\varepsilon_k \leq 1/K$ and the bound in Theorem 2 reduces to (5) for $K \geq 2$.

5.2 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 $h^i = (h_1^i, \dots, h_K^i)$. At time t , the hash tables are updated as follows. Let $x^{(t)}$ be the observation. Then for all $k \in [K], i \in [d], y \in [m]$:

$$\begin{aligned} c_k(i, y) &\leftarrow c_k(i, y) + \mathbb{1}\left\{y = h_k^i(x_k^{(t)})\right\}, \\ \bar{c}_k(i, y) &\leftarrow \bar{c}_k(i, y) + \mathbb{1}\left\{y = h_k^i(x_k^{(t)}) + M(x_{\text{pa}(k)}^{(t)} - 1)\right\}. \end{aligned} \quad (7)$$

This update takes $O(Kd)$ time. **GMSketch** maintains d instances of **GMHash**. Therefore, it consumes $O(Kmd)$ space. Now we show that \hat{P} is a good approximation of \bar{P} .

Theorem 3. *Let \hat{P} be the estimator from Algorithm 2. Let h^1, \dots, h^d be d random and pairwise-independent hashes, and m be the number of bins in each hash function. 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 at least $1 - \delta$ probability, where ε_k are defined in Theorem 2 for $\delta = 1/4$.

Proof. The proof is in Appendix. The key idea is the so-called median trick on d estimates of **GMHash** in Theorem 2 for $\delta = 1/4$.

Similarly to Sect. 5.1, 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 .

5.3 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:

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

where $\hat{P}_k(x_k, x_{\text{pa}(k)})$ is the CM sketch of $\bar{P}_k(x_k, x_{\text{pa}(k)})$ 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 in the same way as in (7). This update takes $O(Kd)$ time and **GMFactorSketch** consumes $O(Kmd)$ space. Now we show that \hat{P} is a good approximation of \bar{P} .

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**

$$\hat{P}_k^i(x_k) \leftarrow \frac{c_k(i, h_k^i(x_k))}{n}$$

$$\hat{P}_k(x_k) \leftarrow \min_{i \in [d]} \hat{P}_k^i(x_k)$$

// Count-min sketches for variable-parent pairs in \mathcal{G} **for all** $k = 2, \dots, K$ **do****for all** $i = 1, \dots, d$ **do**

$$\hat{P}_k^i(x_k, x_{\text{pa}(k)}) \leftarrow \frac{\bar{c}_k(i, h_k^i(x_k + M(x_{\text{pa}(k)} - 1)))}{n}$$

$$\hat{P}_k(x_k, x_{\text{pa}(k)}) \leftarrow \min_{i \in [d]} \hat{P}_k^i(x_k, x_{\text{pa}(k)})$$

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

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

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

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

Theorem 4. Let \hat{P} be the estimator from Algorithm 3. Let h^1, \dots, h^d be d random and pairwise-independent hashes, and m be the number of bins in each hash function. 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 at least $1 - \delta$ probability, where:

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

Proof. The proof is in Appendix. The main idea of the proof is similar to that of Theorem 2. The key difference is that we prove that event (6) is unlikely for any $k \in [K] - \{1\}$ by bounding the probabilities of events:

$$\begin{aligned} \hat{P}_{\text{pa}(k)}(x_{\text{pa}(k)}) - \bar{P}_{\text{pa}(k)}(x_{\text{pa}(k)}) &> \varepsilon_k \alpha_k, \\ \hat{P}_k(x_k, x_{\text{pa}(k)}) - \bar{P}_k(x_k, x_{\text{pa}(k)}) &> \varepsilon_k \alpha_k, \end{aligned}$$

where $\hat{P}_k(x_k, x_{\text{pa}(k)})$ is the CM sketch of $\bar{P}_k(x_k, x_{\text{pa}(k)})$ and $\hat{P}_{\text{pa}(k)}(x_{\text{pa}(k)})$ is the CM sketch of $\bar{P}_{\text{pa}(k)}(x_{\text{pa}(k)})$.

As in Sects. 5.1 and 5.2, 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 .

5.4 Lower Bound

Our bounds depend on query-specific constants $\bar{P}_k(x_k, x_{\text{pa}(k)})$, which can be small. We argue that this dependence is intrinsic. In particular, we show that there exists a family of distributions \mathcal{C} such that any data structure that can summarize any $\bar{P} \in \mathcal{C}$ well must consume $\Omega(\Delta^{-1}(\mathcal{C}))$ space, where:

$$\Delta(\mathcal{C}) = \min_{\bar{P} \in \mathcal{C}, x \in [M]^K, k \in [K] - \{1\}: \bar{P}(x) > 0} \bar{P}_k(x_k, x_{\text{pa}(k)}).$$

Our family of distributions \mathcal{C} is defined on two dependent random variables, where X_1 is the parent and X_2 is its child. Let m be an integer such that $m = 1/\epsilon$ for some fixed $\epsilon \in [0, 1]$. Each model in \mathcal{C} is defined as follows. The probability of any m values of X_1 is ϵ . The conditional of X_2 is defined as follows. When $\bar{P}_1(i) > 0$, the probability of any m values of X_2 is ϵ . When $\bar{P}_1(i) = 0$, the probability of all values of X_2 is $1/M$. Note that each model induces a different distribution and that the number of the distributions is $\binom{M}{m}^{m+1}$, because there are $\binom{M}{m}$ different priors \bar{P}_1 and $\binom{M}{m}$ different conditionals $\bar{P}_2(\cdot | i)$, one for each $\bar{P}_1(i) > 0$. We also note that $\Delta(\mathcal{C}) = \epsilon^2$. The main result of this section is proved below.

Theorem 5. *Any data structure that can summarize any $\bar{P} \in \mathcal{C}$ as \hat{P} such that $|\hat{P}(x) - \bar{P}(x)| < \epsilon^2/2$ for any $x \in [M]^K$ must consume $\Omega(\Delta^{-1}(\mathcal{C}))$ space.*

Proof. Suppose that a data structure can summarize any $\bar{P} \in \mathcal{C}$ as \hat{P} such that $|\hat{P}(x) - \bar{P}(x)| < \epsilon^2/2$ for any $x \in [M]^K$. Then the data structure must be able to distinguish between any two $\bar{P} \in \mathcal{C}$, since $\bar{P}(x) \in \{0, \epsilon^2\}$. At the minimum, such a data structure must be able to represent the index of any $\bar{P} \in \mathcal{C}$, which cannot be done in less than:

$$\log_2 \left(\binom{M}{m}^{m+1} \right) \geq \log_2 \left((M/m)^{m^2+m} \right) \geq m^2 \log_2(M/m)$$

bits because the number of distributions in \mathcal{C} is $\binom{M}{m}^{m+1}$. Now note that $m^2 = 1/\epsilon^2 = \Delta^{-1}(\mathcal{C})$.

It is easy to verify that **GMFactorSketch** is such a data structure for $m = 5e\Delta^{-1}(\mathcal{C})$ in Theorem 4. In this setting, **GMFactorSketch** consumes $O(\log(1/\delta)\Delta^{-1}(\mathcal{C}))$ space. The only major difference from Theorem 5 is that **GMFactorSketch** makes a mistake with at most δ probability. Up to this factor, our analysis is order-optimal and we conclude that the dependence on the reciprocal of $\min_{k \in [K] - \{1\}} \bar{P}_k(x_k, x_{\text{pa}(k)})$ cannot be avoided in general.

6 Comparison with the Count-Min Sketch

In general, the error bounds in Theorems 1 and 4 are not comparable, 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 n observations such that $\bar{P} = \tilde{P}$. This holds, for instance, when $n \rightarrow \infty$, 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 this section, we construct a class of graphical models where **GMFactorSketch** has a tighter error bound than the CM sketch. This class contains naive Bayes models with $K + 1$ variables:

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

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, for any x with at least $1 - \delta$ probability:

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

where m is the number of hashing bins in **GMFactorSketch**. Since $\hat{P}_1 = P_1$, we can omit $1 + \varepsilon_1$ from Theorem 4. This approximation consumes, up to logarithmic factors in K , $2Km \log(1/\delta)$ space. The CM sketch (Sect. 2.1) guarantees that:

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

for any x with at least $1 - \delta$ probability, where m' is the number of hashing 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:

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

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

Roughly speaking, the bound in (9) seems to be tighter than that in (11) because it contains 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 a single $1/m$. Now we prove this claim formally. Before we start, note that both upper bounds in (9) and (11) contain $\frac{1}{2} \left[\prod_{k=2}^{K+1} C_k \right]$. Therefore, we can divide both bounds by this constant and get that the upper bound in (9) is tighter than that in (11) when:

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

Now we rewrite each $(1 + 2e/(C_k m))$ 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) happens when:

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

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:

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

for $m \geq 4eK$. This 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 6. *Let P be the distribution in (8) and x be such that $P_k(x_k | x_1) \leq 1/2$ 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:

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

By the pidgeonhole principle, Theorem 6 guarantees improvements in at least $2(M-1)^K$ points x in any distribution in (8). We can bound the fraction of these points from below as:

$$\frac{2(M-1)^K}{2M^K} = \exp[K \log(M-1) - K \log M] \geq \exp\left[-\frac{K}{M-1}\right] \geq 1 - \frac{K}{M-1}.$$

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

7 Experiments

In this section, we compare our algorithms (Sect. 5) and the CM sketch on the synthetic problem in Sect. 6, and also on a real-world problem in online advertising.

7.1 Synthetic Problem

We experiment with the naive Bayes model in (8), where $P_1(1) = P_1(2) = 0.5$; and:

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

for any $k \in [K+1] - \{1\}$ and $N \ll M$. The model defines the following distribution over $x = (x_1, \dots, x_K)$: when $x_1 = 1$, $P(x) = 0.5N^{-K}$ and we refer to the example x as *heavy*; and when $x_1 = 2$, $P(x) = 0.5(M-N)^{-K}$ and we refer to the example x as *light*. The heavy examples are much more probable 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. 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 $\tilde{P} \rightarrow P$ and $\hat{P} \rightarrow P$, and this is a fair way of comparing our methods to the CM sketch. We choose $d = 5$. We observe similar trends for other values of d . All results are averaged over 20 runs.

7.2 Easy Synthetic Problem

We choose $K = 4$ and $N = 8$, and then $P(x) = 2^{-13}$ for all 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 this problem is *easy for the CM sketch*.

Our results are reported in Fig. 1a. We observe that all of our algorithms outperform the CM sketch. In particular, note that P_{CM} approximates P well for any heavy x in about 2^{15} space. Our algorithms achieve the same quality of the approximation in at most 2^{13} space. **GMFactorSketch** consumes 2^{10} space, which is almost two orders of magnitude less than the CM sketch.

7.3 Hard Synthetic Problem

We set $K = 32$ and $N = 64$, and then $P(x) = 2^{-193}$ for all 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 this problem is *hard for the CM sketch*.

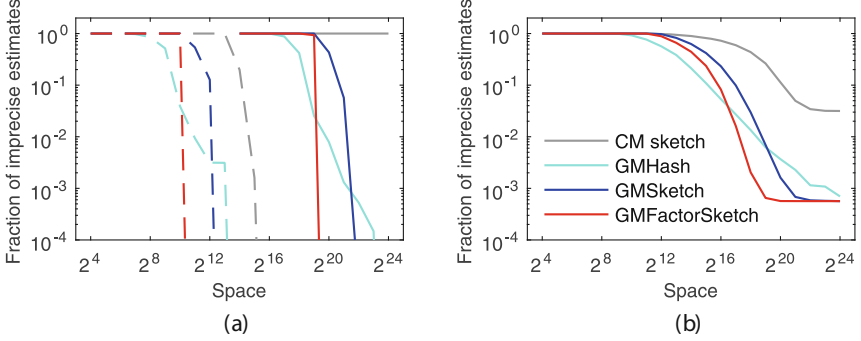


Fig. 1. **a.** Evaluation of the CM sketch, GMHash, GMSketch, and GMFactorSketch on the easy problem in Sect. 7.2 (dashed lines) and the hard problem in Sect. 7.3 (solid lines). **b.** Evaluation on the real-world problem in Sect. 7.4.

Our results are reported in Fig. 1a and we observe three major trends. First, the CM sketch performs poorly. Second, as in Sect. 7.2, 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.

7.4 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. We experiment with two months of data of a medium-sized customer of *Adobe Marketing Cloud*². This is 65M page views, each of which is described by six variables: COUNTRY, CITY, PAGE NAME, STARTING PAGE NAME, CAMPAIGN, and BROWSER. Variable PAGE NAME takes on more than 42k values and has the highest cardinality. We approximate the distribution P over our variables by a naive Bayes model, where the class variable is $X_1 = \text{COUNTRY}$. Since the behavior of users is often driven by their locations, this approximation is quite reasonable.

All compared algorithms are trained on 1M i.i.d. examples from distribution P and tested on all heavy examples in this sample. We say that the example x is *heavy* when $P(x) > 10^{-6}$. The rest of the setup is identical to that in Sect. 7.1.

² <http://www.adobe.com/marketing-cloud.html>.

Our results are reported in Fig. 1b. We observe the same trends as in Sect. 7.3. The CM sketch performs poorly, and our methods outperform it at the same space for any space from 2^{13} to 2^{24} . Also note that none of the compared methods achieve zero mistakes. This is because our sample size n is not large enough to approximate P well in all heavy x . Even if $\hat{P} = \bar{P}$, our methods would still make mistakes.

8 Conclusions

Structured high-cardinality data arises in many domains. Probability distributions over such data cannot be estimated easily with guarantees by either graphical models [9], a popular approach to reasoning with structured data; or count sketches [17], a common approach to approximating probabilities in high-cardinality streams of data. We bring together the ideas of graphical models and sketches, and propose three approximations to the MLE in graphical models with high-cardinality variables. We analyze them and prove that our best approximation, **GMFactorSketch**, outperforms the CM sketch on a class of naive Bayes models. We validate these findings empirically.

The MLE is a common approach to estimating the parameters of graphical models [9]. We propose, analyze, and empirically evaluate multiple space-efficient approximations to this procedure with high-cardinality variables. In this work, we focus solely on the problem of estimating $\bar{P}(x)$, the probability at a single point x . However, note that our models are constructed from Bayesian networks, which can answer $P(Y = y)$ for any subset of variables Y with values y . We do not analyze such inference queries and leave this for future work.

Our work is the first formal investigation of approximations on the intersection of graphical models and sketches. One of our key results is that **GMFactorSketch** yields a constant-factor multiplicative approximation to $\bar{P}(x)$ for any x with probability of at least $1 - \delta$ in $O(K^2 \log(K/\delta) \Delta^{-1}(x))$ space, where K is the number of variables and $\Delta(x)$ reflects the hardness of query x . This result is encouraging because the space is only quadratic in K and logarithmic in $1/\delta$. The space also depends on constant $\Delta(x)$, which can be small. This constant is intrinsic (Sect. 5.4); and this indicates that the problem of approximating $\bar{P}(x)$ well, for any \bar{P} and x , is intrinsically hard.

References

1. Belle, V., Van den Broeck, G., Passerini, A.: Hashing-based approximate probabilistic inference in hybrid domains. In: Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence (2015)
2. Boutilier, C., Friedman, N., Goldszmidt, M., Koller, D.: Context-specific independence in Bayesian networks. In: Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence, pp. 115–123 (1996)
3. Charikar, M., Chen, K., Farach-Colton, M.: Finding frequent items in data streams. Theor. Comput. Sci. **312**(1), 3–15 (2004)

4. Cormode, G., Muthukrishnan, S.: An improved data stream summary: The count-min sketch and its applications. *J. Algorithms* **55**(1), 58–75 (2005)
5. Cormode, G., Muthukrishnan, S.: What's hot and what's not: Tracking most frequent items dynamically. *ACM Trans. Database Syst.* **30**(1), 249–278 (2005)
6. Ermon, S., Gomes, C., Sabharwal, A., Selman, B.: Taming the curse of dimensionality: Discrete integration by hashing and optimization. In: *Proceedings of the 30th International Conference on Machine Learning*, pp. 334–342 (2013)
7. Flajolet, P., Fusy, E., Gandouet, O., Meunier, F.: Hyperloglog: the analysis of a near-optimal cardinality estimation algorithm. In: *Proceedings of the 2007 Conference on Analysis of Algorithms*, pp. 127–146 (2007)
8. Flajolet, P., Martin, G.N.: Probabilistic counting algorithms for data base applications. *J. Comput. Syst. Sci.* **31**(2), 182–209 (1985)
9. Jensen, F.: *Introduction to Bayesian Networks*. Springer, Heidelberg (1996)
10. Koller, D., Friedman, N.: *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, Cambridge (2009)
11. Lafferty, J., McCallum, A., Pereira, F.: *Conditional random fields: Probabilistic models for segmenting and labeling sequence data* (2001)
12. Liberty, E.: Simple and deterministic matrix sketching. In: *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 581–588 (2013)
13. Matushevych, S., Smola, A., Ahmed, A.: Hokusai - Sketching streams in real time. In: *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence* (2012)
14. McGregor, A., Vu, H.: Evaluating Bayesian networks via data streams. In: *Proceedings of the 21st International Conference on Computing and Combinatorics*, pp. 731–743 (2015)
15. Misra, J., Gries, D.: Finding repeated elements. *Sci. Comput. Program.* **2**(2), 143–152 (1982)
16. Murphy, K., Torralba, A., Freeman, W.: Using the forest to see the trees: a graphical model relating features, objects, and scenes. *Adv. Neural Inf. Process. Syst.* **16**, 1499–1506 (2004)
17. Muthukrishnan, S.: Data streams: algorithms and applications. *Found. Trend. Theor. Comput. Sci.* **1**(2), 117–236 (2005)
18. Woodruff, D.: Low rank approximation lower bounds in row-update streams. *Adv. Neural Inf. Process. Syst.* **27**, 1781–1789 (2014)