Efficient Methods for Lifted Inference with Aggregate Factors

Jaesik Choi
Computer Science Department
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA

Rodrigo de Salvo Braz and **Hung H. Bui**
Artificial Intelligence Center
SRI International
Menlo Park, CA 94025, USA

Abstract

Aggregate factors (that is, those based on aggregate functions such as *SUM*, *AVERAGE*, *AND* etc) in probabilistic relational models can compactly represent dependencies among a large number of relational random variables. However, propositional inference on a factor aggregating n k-valued random variables into an r-valued result random variable is $O(rk2^n)$. Lifted methods can ameliorate this to $O(rn^k)$ in general and $O(rk \log n)$ for commutative associative aggregators. In this paper, we propose (a) an *exact* solution *constant* in n when k=2 for certain aggregate operations such as *AND*, *OR* and *SUM*, and (b) a close approximation for inference with aggregate factors with time complexity *constant* in n. This approximate inference involves an analytical solution for some operations when k>2. The approximation is based on the fact that the typically used aggregate functions can be represented by linear constraints in the standard $(k-1)$-simplex in $\mathbb{R}^k$ where k is the number of possible values for random variables. This includes even aggregate functions that are commutative but not associative (e.g., the *MODE* operator that chooses the most frequent value). Our algorithm takes polynomial time in k (which is only 2 for binary variables) regardless of r and n, and the error decreases as n increases. Therefore, for most applications (in which a close approximation suffices) our algorithm is a much more efficient solution than existing algorithms. We present experimental results supporting these claims. We also present a (c) third contribution which further optimizes aggregations over multiple groups of random variables with distinct distributions.

1 Introduction

Relational models can compactly (that is, intensionally) represent graphical models involving a large number of random variables, each of them representing a relation between objects in a domain (Koller and Pfeffer 1997; Getoor et al. 2001; Milch et al. 2005; Richardson and Domingos 2006).

While it is possible to take advantage of compactness only for representation and expand the model into a propositional (extensional) form for inference, lifted inference methods try to keep the representation as compact as possible even during inference, increasing efficiency (Poole 2003; de Salvo Braz, Amir, and Roth 2007; Milch et al. 2008; Singla and Domingos 2008).

The first proposed lifted inference solutions could deal only with factors on a fixed number of random variables. *Aggregate* parametric factors (based on aggregate functions such as *OR*, *MAX*, *AND*, *SUM*, *AVERAGE*, *MODE* and *MEDIAN*), which are defined on a varying, intensionally defined set of random variables, still needed to be treated propositionally, with cost exponential in the number n of random variables. (Kisynski and Poole 2009) introduced lifted methods for aggregate factors that reduce this complexity to $O(rk \log n)$ for commutative associative aggregate functions on n k-valued random variables being aggregated into an r-valued random variable (and even $O(rk)$ for *OR* and *MAX*)[1]. However, for general cases (such as the non-associative function *MODE*), their exact inference method has time $O(rn^k)$, that is, polynomial in n.

The contributions of this paper are threefold. We contribute an *exact* solution *constant* in n when $k = 2$ for aggregate operations *AND*, *OR*, *MAX* and *SUM*. We also present an efficient (*constant* in n) approximate algorithm for inference with aggregate factors, for all typical aggregate functions. The potential of a aggregate factor for a valuation v of a set of random variables depends only on the *histogram* on the distribution of k values in V (in what (Milch et al. 2008) calls a *counting formula*). We show that the typical aggregate functions but for *XOR*[2] can be represented by linear constraints in the space of histograms (a $(k-1)$-simplex). Because aggregate factors' potentials on the space of histograms can be approximated by a normal distribution, we can approximately sums over them (which is the main inference operation) by computing the volume under normal distributions truncated by linear constraints. This holds even for *MODE*, which is commutative but not associative.

This approximation can be computed analytically for all operations on binary random variables and for certain operations on multivalued (k>2) random variables such as *SUM* and *MEDIAN*. Otherwise, it is computed by Gibbs sam-

[1] Note that r=n for aggregate functions such as *SUM* of n binaray variables.

[2] *XOR* has its own simple solution.

pling with a limited number of iterations (Geweke 1991; Damien and Walker 2001). Finally, a third contribution is a further optimization for aggregations of multiple groups of random variables, each with its own distribution.

This paper is organized as follows. Section 2 defines relational models and our inference problem, *AFM* (Aggregation Factor Marginalization). Section 3 presents our lifted inference methods for aggregate factors followed by an extended algorithm for the generalized problems in Section 4. Section 5 provides the error bounds of the approximations. We present some empirical results in Section 6. We conclude in Section 7.

2 Background and Problem Definition

We are interested in inference problems over relational models with aggregate factors. We now revisit these concepts.

2.1 First-order Probabilistic Models

A **factor** f is a pair (A_f, ϕ_f) where A_f is a tuple of random variables and ϕ_f is a **potential function** from the range of A_f to the nonnegative real numbers. Given a **valuation** v of random variables (**rv**s), the **potential** of f on v is $w_f(v) = \phi_f(A_f)$.

The joint probability defined by a set F of factors on a valuation v of random variables is the normalization of $\prod_{f \in F} w_f(v)$. If each factor in F is a conditional probability of a child random variable given the value of its parent random variables, and there are no directed cycles in the graph formed by directed edges from parents to children, then the model defines a Bayesian network. Otherwise it is an undirected model.

We can have parameterized (indexed) random variables by using **predicates**, which are functions mapping parameter values (indices) to random variables. A **relational atom** is an application of a predicate, possibly with free variables. For example, a predicate $friends$ is used in atoms $friends(X, Y)$, $friends(X, bob)$ and $friends(john, bob)$, where X and Y are free variables and $john$ and bob possible parameter values. $friends(john, bob)$ is a **ground atom** and directly corresponds to a random variable.

A **parfactor** is a tuple (L, C, A, ϕ) composed of a set of parameters (also called *logical variables*) L, a constraint C on L, a tuple of atoms A, and a potential function ϕ. Let a **substitution** θ be an assignment to L and $A\theta$ the relational atom (possibly ground) resulting from replacing logical variables by their values in θ. A parfactor g stands for the set of factors $gr(g)$ with elements $(A\theta, \phi)$ for every assignment θ to the parameters L that satisfies the constraint C. A First-order Probabilistic Model (**FOPM**) is a compact, or intensional, representation of a graphical model. It is composed by a **domain**, which is the set of possible parameter values (referred to as **domain objects**) and a set of parfactors. The corresponding graphical model is the one defined by all instantiated factors. The joint probability of a valuation v according to a set of parfactors G is

$$P(v) = 1/Z \prod_{g \in G} \prod_{f \in gr(g)} w_f(v), \qquad (1)$$

where Z is a normalization constant.

Example: The dependence between political ads and votes in the example in Figure 1 can be compactly represented by the parfactor $(\{i\}, \top, (V(i), Ads), P(V(i)|Ads))$ with a domain formed by the set of voters ($\top$ represents a tautology, so no constraints are posed on i and instances are generated for all voters). The figure uses the more traditional notation V_i, equivalent to $V(i)$.

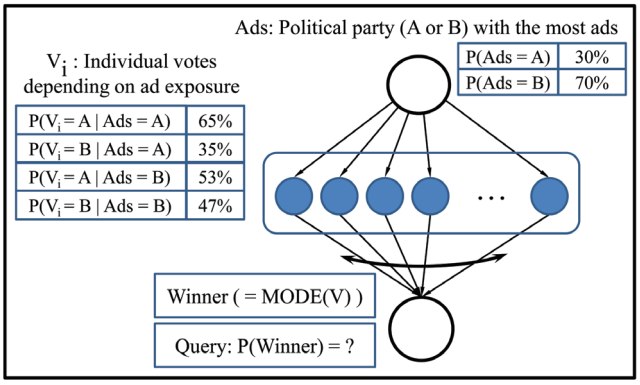

Figure 1: Graphical model on the domain of the election of one of two parties A and B. The random variable Ads indicates which party has the most ads in the media. The variables V_i indicate the vote of each person in a population, modeled as a dependence of ad exposure. The $Winner$ variable indicates the winner and it is determined by the majority (*MODE*) of votes. We would like to estimate the probability of each party winning the election given this model.

.

2.2 Aggregate Factors and Parfactors

An **aggregate factor** is a factor $((X_1, \ldots, X_n, Y, \phi_\otimes))$ where $\phi_\otimes$ establishes that the valuation y of Y must be the result of an aggregation function $\otimes$ over the valuation $x_1, \ldots, x_n$ of $X_1, \ldots, X_n$:

$$\phi_\otimes(x_1, \ldots, x_n, y) = \begin{cases} 1 & \text{if } y = \bigotimes_{i=1,\ldots,n} x_i \\ 0 & \text{otherwise} \end{cases} . \qquad (2)$$

We consider the aggregate functions *OR*, *MAX*, *AND*, *XOR*, *SUM*, *AVERAGE*, *MODE* and *MEDIAN*. Noisy versions such as *Noisy-OR* can be represented by adding an extra factor on x_i.[3]

An **aggregate parfactor** $g = (L, C, X, \otimes, Y)$, where X and Y are now relational atoms, can be used by FOPMs to compactly represent a set of aggregate factors. The set $gr(g)$ of ground factors instantiated from g comprises the aggregate factors $((X\theta_0\theta_1, \ldots, X\theta_0\theta_n, Y\theta_0), \phi_\otimes)$, for each substitution θ_0 on the logical variables in Y consistent with constraint C, and substitutions $\theta_1, \ldots, \theta_n$ on the logical variables in X but not in Y consistent with C. For the example in Figure 1, the conditional probability of $Winner$

[3]Our definitions are based on (Kisynski and Poole 2009) but differ from theirs in this aspect; while our aggregate factors are deterministic, theirs include an extra potential for noisy versions. As explained, we can do the same with an extra factor/parfactor.

can be compactly represented by the aggregate parfactor $(i, \top, V(i), MODE, Winner)$. More general aggregation cases (for example, with aggregated random variables sets including more than one predicate) can be normalized to this type of aggregated parfactor, as detailed in (Kisynski and Poole 2009).

2.3 Inference with Aggregate Parfactors

We are interested in the inference problem of marginalizing a set of rvs in an FOPM with aggregate factors to determine the marginal density of others. As shown by (Kisynski and Poole 2009), this can be done by using C-FOVE (Milch et al. 2008) extended with a lifted operation for summing random variables out of an aggregate parfactor. These summations can be reduced to the Aggregate Factor Marginalization (AFM) calculation:

$$\phi'_{\mathbf{y}}(y) = \sum_{x_1,,x_n} \left(\phi_{\otimes}(y, x_1, \ldots, x_n) \prod_{1 \le i \le n} \phi_{\mathbf{x}}(x_i) \right).$$

where $\phi_{\mathbf{x}}$ is the (same for all i) potential product of all other factors in the model that have X_i as an argument, and $\phi'_{\mathbf{y}}$ is the resulting potential on y alone. This subproblem is also one that needs to be solved in extending Lifted Belief Propagation (Singla and Domingos 2008) to deal with aggregate factors.

(Kisynski and Poole 2009) shows how, when different x_i have different potential functions on them, the problem can be normalized (by splitting and using auxiliary variables) to multiple such sums in which this uniformity holds. Similarly, we can separate the case in which only *some* x_i need to be summed out into two different aggregate parfactors, one for all aggregate random variables being summed out, and another for the remaining ones.

A direct computation of **AFM** is exponential in n. (Kisynski and Poole 2009) shows lifted operations that can be done in time polynomial or logarithmic in n (depending on certain conditions explained below). In Section 3 we present two lifted methods, one exact and one approximate, with time constant in n.

2.4 Inference Problems with Inequality

We define aggregate factors with inequality constraints by using

$$\phi_{\otimes_\le}(y, x_1, \ldots, x_n) = \begin{cases} 1 & \text{if } y \le x_1 \otimes \cdots \otimes x_n \\ 0 & \text{otherwise} \end{cases}$$

with the corresponding problem **AFM**$[\le]$ defined as

$$\sum_{x_1,\ldots,x_n} \left(\phi_{\otimes_\le}(y, x_1, \ldots, x_n) \cdot \prod_{1 \le i \le n} \phi_{\mathbf{x}}(x_i) \right).$$

$\phi_{\otimes_\ge}$ and **AFM**$[\ge]$ are defined analogously.

2.5 Existing Methods for AFM Problems

MAX and its special case *OR* (as well as their noisy versions) allow factorizations leading to lifted marginalization constant in n (Díez and Galán 2003). These operators can be decomposed into the product of n potentials:[4]

$$\begin{aligned} &\sum_{x_1,\ldots,x_n} \phi_{\otimes}(y, x_1, \ldots, x_n) \cdot \prod_{i=1}^{n} \phi_{\mathbf{x}}(x_i) \\ &= \sum_{y'} \sum_{x_1,\ldots,x_n} \prod_{i=1}^{n} \phi_{\mathbf{y}',\mathbf{y}}(y', y) \cdot \phi_{\mathbf{y}',\mathbf{x}}(y', x_i) \\ &= \sum_{y'} \left(\phi_{\mathbf{y}',\mathbf{y}}(y', y) \prod_{i=1}^{n} \sum_{x_i} \phi_{\mathbf{y}',\mathbf{x}}(y', x_i) \right). \end{aligned} \quad (3)$$

Because the product is over a term independent of n, we can compute it once and exponentiate in time constant in n:

$$= \sum_{y'} \left(\phi_{\mathbf{y}',\mathbf{y}}(y', y) \left(\sum_{x'} \phi_{\mathbf{y}',\mathbf{x}}(y', x') \right)^n \right).$$

For other aggregate functions that happen to be commutative and associative, **AFM** can be computed by a recursive decomposition (Kisynski and Poole 2009) into a subproblem with half the number of aggregated random variables, and therefore in time $O(r^2 k \log n)$ when n is a power of 2:

$$\begin{aligned} &\sum_{x_1,\ldots,x_n} \phi_{\otimes}(y, x_1, \ldots, x_n) \cdot \prod_{i=1}^{n} \phi_{\mathbf{x}}(x_i) \\ &= \sum_{y=y' \otimes y''} \left(\sum_{x_1,\ldots,x_{\frac{n}{2}}} \phi_{\otimes}(y', x_1, \ldots, x_{\frac{n}{2}}) \cdot \prod_{i=1}^{\frac{n}{2}} \phi_{\mathbf{x}}(x_i) \right) \\ &\cdot \left(\sum_{x_{\frac{n}{2}+1},\ldots,x_n} \phi_{\otimes}(y'', x_{\frac{n}{2}+1}, \ldots, x_n) \cdot \prod_{i=\frac{n}{2}+1}^{n} \phi_{\mathbf{x}}(x_i) \right), \end{aligned}$$

where $\phi_{\otimes}(y, x_i) = \begin{cases} 1 & \text{if } y = x_i \\ 0 & \text{otherwise} \end{cases}$.

Note that the two decomposition halves are the same problem up to variable renaming and thus computed in time $O(k \log n)$, r^2 times (once per value of y' or y'' and another per value of y). (Kisynski and Poole 2009) describes the minor adjustments needed when n is not a power of 2.

3 Efficient Methods for AFM Problems

We now present our solutions for **AFM** problems. The exact solutions presented in the previous section are efficient. However, their applicability is limited to some operations (Díez and Galán 2003), or their computational complexity still depends on the number of rvs (Kisynski and Poole 2009). Here, we propose an exact solution for some cases, and new efficient approximate marginalizations that are applicable to more aggregate functions.

3.1 Normal Distribution with Linear Constraints

(Kisynski and Poole 2009) shows how the potential of an aggregate parfactor depends only on the value histogram of its aggregated random variables (histograms were introduced in Counting Elimination (de Salvo Braz, Amir, and Roth 2007) and used as counting formulas in (Milch et al. 2008)).

[4] See (Díez and Galán 2003) for details on $\phi_{\mathbf{y}',\mathbf{y}}$ and $\phi_{\mathbf{y}',\mathbf{x}}$.

Given values $x_1, \ldots, x_n$ for n rvs with the same range, the value histogram of x is a vector h with $h_u = |\{i : x_i = u\}|$ for each u in the rvs' range. When a potential function on $x_1, \ldots, x_n$ depends on the histogram alone, as in the case of aggregate factors, then there is a function $\phi_{\mathbf{h}}$ on histograms such that $\phi(y, x_1, \ldots, x_n) = \phi_{\mathbf{h}}(y, h)$ and $\phi_{\bigotimes}(y, x_1, \ldots, x_n) = \phi_{\bigotimes \mathbf{h}}(y, h)$. In what follows, we describe the binomial case (range of x_i equal to 2) for clarity, but it applies to the multinomial case as well. We can write

$$\sum_{x_1,\ldots,x_n} \phi(y, x_1, \ldots, x_n) \prod_i \phi_{\mathbf{x}}(x_i) = \sum_h \binom{n}{h_1} \phi_{\mathbf{h}}(y, h) p_1^{h_1} p_0^{n-h_1}, \quad (4)$$

where p_0, p_1 are the normalizations of $\phi_{\mathbf{x}}$. This corresponds to grouping assignments on x into their corresponding histograms h, and iterating over the histograms (which are exponentially less many), taking into account that each histogram corresponds to $\binom{n}{h_1}$ assignments.

We now observe that functions $\phi_{\mathbf{h}}(y, h)$ coming from aggregate factors always evaluate to 0 or 1. Moreover, the set of histograms for which they evaluate to 1 can be described by linear constraints on the histogram components. For example, $\phi_{MODE}(y, h)$ will only be 1 if $h_y \geq h_{y'}$ for all $y' \neq y$. Given $\phi_{\mathbf{h}}$ and y, let C_y be the set of histograms h such that $\phi_{\mathbf{h}}(y, h)=1$. Then (4) can be rewritten as

$$\sum_{h \in C_y} \binom{n}{h_1} p_1^{h_1} p_0^{n-h_1},$$

which is the probability of a set of h_1 values under a binomial distribution. For large n, according to the Central Limit Theorem (Rice 2006), the binomial distribution is approximated by the normal distribution $N(np_1, np_1p_0)$ with density function f. Then

$$\sum_{h \in C_y} \binom{n}{h_1} p_1^{h_1} p_0^{n-h_1} \approx \int_{h' \in C'_y} f(h')\, dh',$$

where C'_y is a continuous region in the $(k-1)$-simplex corresponding to C_y (which is defined in discrete space). Table 1 lists C_y and an appropriate C'_y for the several aggregate factor potentials, for both **AFM** and **AFM**$[\geq]$.

Let's see two examples. For **AFM** on *MODE* on binary variables, $y = 1$, and histograms with $h(1) = t$, C_y is $h_1 \geq h_0$ and C'_y is $t \in \left[\lfloor \frac{n}{2} \rfloor + 0.5, n + 0.5\right]$[5], so we compute

$$\int_{t=\lfloor \frac{n}{2} \rfloor + 0.5}^{n+0.5} f(t)\, dt,$$

which can be done in constant time. Let us also consider **AFM** and **AFM**$[\geq]$ on *SUM* with n=100 rvs representing ratings of 100 people who watch a movie. Each person gives ratings of either 0 (negative) or 1 (positive), with probabilities 0.55 and 0.45, respectively (p_0=0.55). We are interested in the summation of those votes (r=100). Figure 2 shows the probability density of the number of positive ratings. The bars in red in (a) and (b) panels show the area corresponding to the result for **AFM** and **AFM**$[\geq]$, respectively, for y=50. The former can have the exact binomial distribution form computed in constant time, while the latter can have the normal distribution approximation computed in constant time. Therefore, the marginal on Y can be approximated in $O(r)$. (Kisynski and Poole 2009)'s algorithm, on the other hand, takes $O(r \log n)$, and (Díez and Galán 2003) is not applicable.

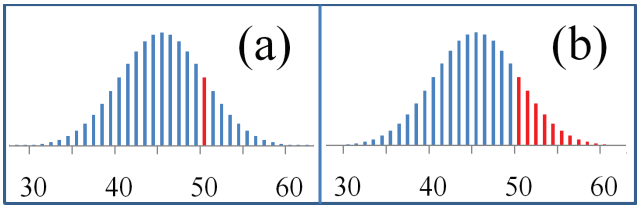

Figure 2: Histogram with a binomial distribution with (a) equality and (b) inequality constraints.

We now explain the method in more detail for two different cases: aggregated binary random variables (k=2), which can be dealt with analytically, and aggregated multivalued random variables (k>2).

3.2 Binary Variables Case

AFM Problem For *AND*, *OR*, *MIN*, *MAX* and *SUM*, an exact solution with time constant in n for **AFM** for the binary case can be computed, for the appropriate choices of p_0 and p_1, as

$$\phi'_{\mathbf{y}}(y) = \binom{n}{y} p_0^{n-y} \cdot {p_1}^y.$$

AVERAGE can be solved by using $\phi'_{\mathbf{y}}$ obtained from *SUM* on y/n. This solution follows from the fact that, for the above cases, one needs the potential of a single histogram.

For *MODE* and *MEDIAN*, exact solutions for **AFM** are of the following form, with time linear in n:

$$\phi'_{\mathbf{y}}(TRUE) = \sum_{i=\lfloor \frac{n}{2} \rfloor + 1}^{n} \binom{n}{i} p_0^{n-i} \cdot {p_1}^i.$$

Such solutions are more expensive because they measure the density of a region of histograms. They can be approximated by the Normal distribution in the following way:

$$\phi'_{\mathbf{y}}(TRUE) \approx \int_{t=\lfloor \frac{n}{2} \rfloor + 0.5}^{n+0.5} \frac{exp\left(-\frac{(t-np_1)^2}{2 \cdot np_1(1-p_1)}\right)}{\sqrt{2\pi \cdot np_1(1-p_1)}}\, dt.$$

Note that *MODE* is not solved by either (Díez and Galán 2003)'s factorization or (Kisynski and Poole 2009)'s logarithmic algorithm, while our approach can compute an approximation in constant time. For n is 100, $p_1 = 0.45$, the

[5]Here, $+0.5$ and -0.5 are continuity corrections for accurate approximations.

Operator	Problem	y	C_y	C'_y
AND	**AFM**	$TRUE$	$h_{TRUE} = n$	not needed (cheap exact solution)
OR	**AFM**	$FALSE$	$h_{FALSE} = n$	not needed (cheap exact solution)
SUM	**AFM**	y	$\sum_i i \times h_i = y$	$y - 0.5 \leq \sum_i i \times h_i \leq y + 0.5$
SUM	**AFM**$[\geq]$	y	$\sum_i i \times h_i \leq y$	$\sum_i i \times h_i \leq y - 0.5$
MAX	**AFM**	y	$h_y > 0$ and $\forall i{>}y \;\; h_i = 0$	$h_y > 0.5$ and $\forall i{>}y \;\; -0.5 \leq h_i \leq 0.5$
MAX	**AFM**$[\geq]$	y	$\forall i{>}y \;\; h_i = 0$	$\forall i{>}y \;\; -0.5 \leq h_i \leq 0.5$
MODE	**AFM**	y	$\forall i{\neq}y \;\; h_y > h_i$	$\forall i{\neq}y \;\; h_y > h_i$
MEDIAN	**AFM**	y	$\sum_{i=1}^{y-1} h(i) < \frac{n}{2} \leq \sum_{i=y}^{n} h(i)$	$\sum_{i=1}^{y-1} h(i) + 0.5 \leq \lfloor \frac{n}{2} \rfloor \leq \sum_{i=y}^{n} h(i) - 0.5$
MEDIAN	**AFM**$[\geq]$	y	$\sum_{i=1}^{y-1} h(i) \geq \frac{n}{2}$	$\sum_{i=1}^{y-1} h(i) - 0.5 \geq \lfloor \frac{n}{2} \rfloor$

Table 1: Constraints to be used in binomial (multinomial) distribution exact calculations (C_y) and (multivariate) Normal distribution approximations (C'_y). The table does not exhaust all combinations. However those omitted are easily obtained from the presented ones. For example, $\phi_{OR}(T, x) = 1 - \phi_{OR}(F, x)$, $\phi_{AVERAGE}(y, x) = \phi_{SUM}(y \times n, x)$, and $\phi_{MODE\geq}(y, x) = \sum_{y' \leq y} \phi_{MODE}(y', x)$.

exact solution is about 0.18272. Our approximate solution is about 0.18286. Thus, the error is less than 0.1% of the exact solution.

AFM$[\leq]$ and AFM$[\geq]$ Problems For binary aggregated random variables, these problems are different from **AFM** only for the *SUM* (and thus, *AVERAGE*) case. For *SUM* we can use the approximation

$$\phi'_{\mathbf{y}}(y) = \sum_{i=y}^{n} \binom{n}{i} p_1^i (1-p_1)^{n-i} \approx \int_{t=y-0.5}^{n+0.5} \frac{exp\left(-\frac{(t-np_1)^2}{2 \cdot np_1(1-p_1)}\right)}{\sqrt{2\pi \cdot np_1(1-p_1)}}\, dt.$$

3.3 Multivalued Variables Case

In the multivalued ($k{>}2$) case, there is a need to compute the probability of a linearly constrained region of histograms, which motivates us to consider approximate solutions with the multivariate Normal distribution. Consider the following example: suppose that the aggregation function is *SUM*. There are 100 rvs representing ratings of 100 people who watch a movie. Each person gives ratings among 0, 1 and 2 (0 is lowest and 2 is highest). We want to calculate the sum of ratings from 100 people when each person gives a rating 0 with 0.35 ($p(x_i{=}r0){=}0.35$), 1 with 0.35 ($p(x_i{=}r1){=}0.35$), and 2 with 0.3 ($p(x_i{=}r2){=}0.3$). The probability of histograms is provided by the multinomial distribution, as shown in Figure 3. The colored bars in (a) represent the probability of the ratings sum being exactly 100. If instead we wish to determine the probability of the ratings sum exceeding 100, we have an **AFM**$[\geq]$ instance, with a probability corresponding to the colored bars in the (b) panel. In both cases, we need to compute the volume of a histogram region.

As in the previous section, the multinomial distribution can be approximated by the multivariate normal distribution. Suppose that each rv may have three values with probability p_0, p_1 and p_2 ($p_0 + p_1 + p_2 = 1$), respectively. Then the multinomial distribution of h_0, h_1 and h_2 chosen from n rvs is

$$\binom{n}{h_0\, h_1\, h_2} \cdot p_0^{h_0} \cdot p_1^{h_1} \cdot p_2^{h_2} = \frac{n!}{h_0! h_1! h_2!} \cdot p_0^{h_0} \cdot p_1^{h_1} \cdot p_2^{h_2}.$$

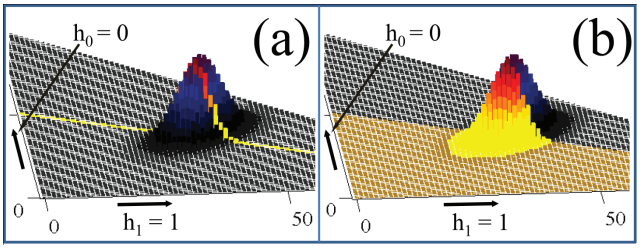

Figure 3: Histogram space for multinomial distributions with (a) equality and (b) inequality constraints.

The corresponding bivariate (i.e. (3-1) multivariate) normal distribution of $\mathbb{X} = [h_0\, h_1]$ chosen from n rvs is as follows (Note that $h_2 = n - h_1 - h_2$),

$$\frac{1}{(2\pi)^{2/2}|\Sigma|^{1/2}} \cdot exp\left(-\frac{1}{2}(\mathbb{X} - \mu)\Sigma^{-1}(\mathbb{X} - \mu)'\right),$$

when the μ and Σ are

$$\mu = [np_0\, np_1],\; \Sigma = \begin{pmatrix} np_0(1-p_0) & np_1p_2 \\ np_2p_1 & np_2(1-p_2) \end{pmatrix}.$$

Analytical Solution for Operators with a Single Linear Constraint As in the previous section, we set p_0, p_1 and p_2 as 0.35, 0.35 and 0.3 respectively and y as 100. Any operator with a single linear constraint (e.g. **AFM**, **AFM**$[\leq]$ and **AFM**$[\geq]$ on *SUM*, and **AFM**$[\leq]$ and **AFM**$[\geq]$ on *MEDIAN*) allows an analytical solution because there is a linear transformation from $\mathbb{X} = [h_0\, h_1]$ to y. Consider the following linear transform $y = 0{\cdot}h_0 + 1{\cdot}h_1 + 2{\cdot}h_2 = 200 - 2{\cdot}h_0 - h_1$. When we represent the transform as $y = A\mathbb{X} + B$, the new distribution of y is given by the 1-D Normal distribution:

$$\frac{1}{\sqrt{2\pi\Sigma_y}} \cdot exp\left(-\frac{(y-\mu_y)^2}{2\Sigma_y}\right),$$

where $\mu_y = A\mu + B$ and $\Sigma_y = A\Sigma A^T$ are scalars. From the transformation the solution of **AFM** for $y{=}100$ can be calculated in the following way:

$$\frac{1}{\sqrt{2\pi\Sigma_y}} \int_{y=100-0.5}^{100+0.5} exp\left(-\frac{(y-\mu_y)^2}{2\Sigma_y}\right) dy.$$

The solutions of **AFM**[$\leq$] and **AFM**[$\geq$] for y=100 can be calculated in similar ways.

Sampling for Remaining Operators In general, integration of a multivariate truncated normal does not allow an analytical solution. Fortunately, efficient **Gibbs sampling** methods (e.g. (Geweke 1991; Damien and Walker 2001)) are applicable to the truncated normal in straightforward ways, even with several linear constraints. This immediately feeds to an approximation with time complexity not depending on n, the number of rvs.

4 Aggregate Factor with Multiple Atoms

We now consider a generalized situation. Previous sections assume that all rvs in a relational atom have the same distribution. Here, we deal with the issue of aggregating J distinct groups of random variables, each represented by a relational atom X_j with n_j groundings and a distinct potential $\phi_{\mathbf{x_j}}$, for $1 \leq j \leq J$.

$$y = \bigotimes_{\substack{1<j<J \\ 1<i<n_j}} x_{j,i}.$$

This problem, **AFM-M**, is an extension of the **AFM**. The **AFM-M** is to calculate a marginal

$$\sum_{x_{1,1},\cdots,x_{J,n_J}} \phi_{\otimes}(y, x_{1,1}, \cdots, x_{J,n_J}) \prod_{j=1}^{J} \prod_{1 \leq i \leq n_j} \phi_{\mathbf{x_j}}(x_{j,i}).$$

One approach is to compute an aggregate y_j^0 per atom j, and then combine each pair y_j^i and y_{j+1}^i into $y_{\lfloor j/2 \rfloor}^{i+1}$ until they are all aggregated. This will have complexity $O(J \log J)$ but works only for associative operators. For non-associative operators, we need to calculate the marginal for each X_j independently:

$$\sum_{h^1,\cdots,h^J} \phi_{\otimes \mathbf{h}}(y,\mathbf{h}) \left(\binom{n_1}{h_1^1} p_{1,0}^{h_0^1} p_{1,1}^{h_1^1} \cdots \binom{n_J}{h_1^J} p_{J,0}^{h_0^J} p_{J,1}^{h_1^J} \right),$$

where $p_{j,0}$ and $p_{j,1}$ are the normalization of $\phi_{\mathbf{x_j}}(0)$ and $\phi_{\mathbf{x_j}}(1)$; h^j is a histogram for atom j, and $\mathbf{h}$ is the combined histogram. The complexity of this approach is $O(exp(J))$.

Another approach is to make use of the representation of the aggregation operator as a set of linear constraints (Table 1). Note that h^j is approximately Normal when n_j is large, and h^i and h^j are independent when $i \neq j$. Thus, the all-group histogram vector $\mathbf{h}$ is also approximately Normal distributed because it is the Normal sum ($\mathbf{h}_i = \sum_j h_i^j$).

Any linear constraint in Table 1 can be re-expressed as a linear constraint using elements of $\mathbf{h}$, and the multinomial-Normal approximation can be used to yield a similar approximate solution in time constant n, the total number of rvs.

For example, for binary random variables, the Normal approximation of the all-group histogram is:

$$N\left(\sum_{j=1}^{J} n_j p_{j,1}, \sum_{j=1}^{J} n_j p_{j,1} p_{j,0} \right).$$

This way, the time complexity is only $O(J)$ instead of $O(J \log J)$ (or $O(exp(J))$ for non-associative operators).

5 Error Analysis

Here, we discuss error bounds for the multinomial-Normal approximations. In general, the Berry-Esseen theorem (Esseen 1942) gives an upper bound on the error. Suppose that $\phi_{\mathbf{y}}(y)$ and $\widetilde{\phi}_{\mathbf{y}}(y)$ represent the probability mass of a binomial distribution and density of its normal approximation, respectively. Furthermore, we represent the cumulative probabilties as $\Phi_{\mathbf{y}}(y)$ and $\widetilde{\Phi}_{\mathbf{y}}(y)$[6]. Then, given any y, the error between the two cumulative probabilities is bounded (Esseen 1942):

$$\left| \Phi_{\mathbf{y}}(y) - \widetilde{\Phi}_{\mathbf{y}}(y) \right| < c \cdot \frac{p^2 + (1-p)^2}{\sqrt{np(1-p)}},$$

where c is a small (< 1) constant. Thus, the asymptotic error bound is $O(1/\sqrt{n})$, and this extends to probability on any interval.

For k-valued multinomials, suppose that $\Phi_{\mathbf{Y}}(A)$ and $\widetilde{\Phi}_{\mathbf{Y}}(A)$ represent the probability of a multinomial distribution and its multivariate normal approximation over a measurable convex set A in R^k. Then, the approximation error is bounded (Gotze 1991):

$$sup_A \left| \Phi_{\mathbf{Y}}(A) - \widetilde{\Phi}_{\mathbf{Y}}(A) \right| < c \cdot \frac{k}{\sqrt{n}},$$

where c depends only on the multinomial parameters and not on n. In our problem, A is determined by linear constraints, hence is convex. Thus, the asymptotic error bound is $O(k/\sqrt{n})$.

6 Experimental Results

We provide experimental results on the example in Figure 1 (which uses the *MODE* aggregate function) which give us an insight on when to use the approximate algorithm instead of the generally applicable exact algorithm based on Counting Formulas (the logarithmic method in (Kisynski and Poole 2009) does not apply to *MODE*).

We compute the utility of any of the methods tested, approximations or exact inference alike, in the following manner. We assume a typical application in which the utility of an error is an inverse quadratic function $U(err) = 1 - err^2$. The utility of a method obtaining error err is normalized by the time t it takes to run, so $U(err, t) = U(err)/t$. For sampling methods, t is the time to convergence. Finally, we plot the *ratio* between the utility of our methods and the utility of the exact inference method.

Therefore, a method is advantageous over the exact inference method when this ratio is greater than 1.

We run an experiment comparing our approximations and the exact inference algorithm for the model in Figure 1. For $k = 2$, we run both the analytical and the sampling method. Given k and n, we randomly choose the potentials, and record the error and the convergence time. Then, we average them over 100 trials to calculate the utility, U_{Approx}.

As shown in Figure 4, our approximate algorithm has much higher utility than the exact method for larger k and n. However, when $k = 2$ (binary variables), the exact method

[6]That is, $\Phi_{\mathbf{y}}(y) = \sum_{i=0}^{y} \phi_{\mathbf{y}}(i)$, and $\widetilde{\Phi}_{\mathbf{y}}(y) = \int_{t=-\infty}^{y} \widetilde{\phi}_{\mathbf{y}}(t)\, dt$.

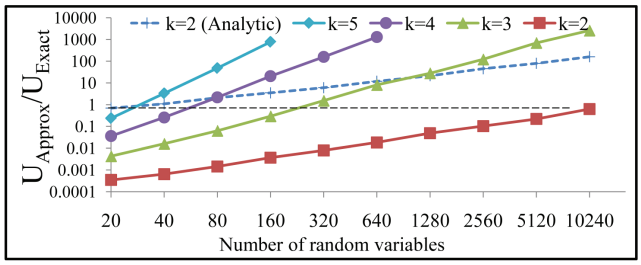

Figure 4: Ratios of utilities of approximate algorithms and exact method (histogram based counting).

has higher utility than sampling for relatively large n (e.g. $n = 10240$). In this case, we can use the efficient analytic integration which applies for $k = 2$. We also show in Figure 5 how the error decreases for different values of k and n.

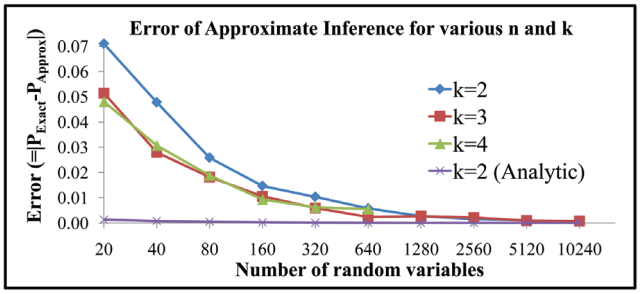

Figure 5: Error curves for different values of k and n.

In addition, we have observed that the convergence time stays flat for various k and n. However, the error of sampling method is noticeable for small n. For example, when $k = 4$, the error is 3.07% with $n = 40$ and 1.82% with $n = 80$. For larger n, this issue is resolved. The error becomes less than 1% when $n = 320$ and negligible when $n > 5120$. These observations are consistent for various k from 2 to 6.

7 Conclusion

Processing aggregate parfactors efficiently is an important problem since they involve functions commonly used in writing models. Our contribution adds efficient exact methods for the binary case k=2, as well as efficient approximations for the cases in which the sets of aggregated variables are large, which is precisely the situation in which we are more likely to use aggregate factors in the first place. It will therefore be an important part of practical applications of relational graphical models.

8 Acknowledgements

We wish to thank Tuyen Ngoc Huynh, David Israel and the anonymous reviewers for their valuable comments.

This material is based upon work supported by the DARPA Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-0181. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of DARPA, the Air Force Research Laboratory (AFRL) or the US government. In the event permission is required, DARPA is authorized to reproduce the copyrighted material for use as an exhibit or handout at DARPA-sponsored events and/or to post the material on the DARPA website.

References

Damien, P., and Walker, S. G. 2001. Sampling truncated normal, beta, and gamma densities. *Journal of Computational and Graphical Statistics* 10(2):206–215.

de Salvo Braz, R.; Amir, E.; and Roth, D. 2007. Lifted first-order probabilistic inference. In Getoor, L., and Taskar, B., eds., *An Introduction to Statistical Relational Learning*. MIT Press. 433–451.

Díez, F. J., and Galán, S. F. 2003. Efficient computation for the noisy MAX. *International Journal of Approximate Reasoning* 18:165–177.

Esseen, C.-G. 1942. On the liapunoff limit of error in the theory of probability. *Arkiv foer Matematik, Astronomi, och Fysik* A28(9):1–19.

Getoor, L.; Friedman, N.; Koller, D.; and Pfeffer, A. 2001. Learning probabilistic relational models. In Džeroski, S., and Lavrac, N., eds., *Relational Data Mining*. Springer-Verlag. 307–335.

Geweke, J. 1991. Efficient simulation from the multivariate normal and student-t distributions subject to linear constraints and the evaluation of constraint probabilities. In *Computer Sciences and Statistics Proceedings the 23rd Symposium on the Interface between*, 571–578.

Gotze, F. 1991. On the rate of convergence in the multivariate clt. *The Annals of Probability* 19(2):724–739.

Kisynski, J., and Poole, D. 2009. Lifted aggregation in directed first-order probabilistic models. In *Proceedings of the 21st international joint conference on Artificial intelligence*, 1922–1929.

Koller, D., and Pfeffer, A. 1997. Object-Oriented Bayesian Networks. In *Proceedings Thirteenth Conference on Uncertainty in Artificial Intelligence*, 302–313.

Milch, B.; Marthi, B.; Russell, S.; Sontag, D.; Ong, D. L.; and Kolobov, A. 2005. BLOG: probabilistic models with unknown objects. In *Proceedings of the 19th international joint conference on Artificial intelligence*, 1352–1359.

Milch, B.; Zettlemoyer, L.; Kersting, K.; Haimes, M.; and Kaelbling, L. P. 2008. Lifted probabilistic inference with counting formulas. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, 1062–1608.

Poole, D. 2003. First-order probabilistic inference. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, 985–991.

Rice, J. A. 2006. *Mathematical Statistics and Data Analysis*. Duxbury Press.

Richardson, M., and Domingos, P. 2006. Markov logic networks. *Machine Learning* 62(1-2):107–136.

Singla, P., and Domingos, P. 2008. Lifted first-order belief propagation. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, 1094–1099.