Multi-Evidence Lifted Message Passing, with Application to PageRank and the Kalman Filter

Babak Ahmadi and **Kristian Kersting**
Knowledge Dicovery, Fraunhofer IAIS
53754 Sankt Augustin, Germany
{firstname.lastname}@iais.fraunhofer.de

Scott Sanner
Statistical ML Group, NICTA & ANU
7 London Circuit, Canberra, Australia
Scott.Sanner@nicta.com.au

Abstract

Lifted message passing algorithms exploit repeated structure within a given graphical model to answer queries efficiently. Given evidence, they construct a lifted network of supernodes and superpotentials corresponding to sets of nodes and potentials that are indistinguishable given the evidence. Recently, efficient algorithms were presented for updating the structure of an existing lifted network with incremental changes to the evidence. In the inference stage, however, current algorithms need to construct a separate lifted network for each evidence case and run a modified message passing algorithm on each lifted network separately. Consequently, symmetries across the inference tasks are not exploited. In this paper, we present a novel lifted message passing technique that exploits symmetries across multiple evidence cases. The benefits of this multi-evidence lifted inference are shown for several important AI tasks such as computing personalized PageRanks and Kalman filters via multi-evidence lifted Gaussian belief propagation.

1 Introduction

Message-passing algorithms, like Belief Propagation (BP) [Pearl, 1991] and its variants, have proved empirically effective for solving hard/computationally intensive problems in a range of important and real-world AI tasks. For example, consider the well-known problem of web page ranking. Viewing the web as a graph where nodes are web pages and directed edges are hyperlinks between web pages, a highly successful web page ranking metric computed from this graph is the PageRank [Brin and Page, 2006]. Converting this graph to an ergodic Markov chain, the PageRank of a web page node v is the (limit) stationary probability that a random walker is at v. Recently, Bickson *et al.* have shown how to compute the PageRank distributively and efficiently using Gaussian belief propagation (GaBP) [2007].

There are many improvements that can be made to (Ga)BP, especially when applied to inference in graphical models containing structural symmetries. Such symmetries are commonly found in first-order and relational probabilistic models (see e.g. Getoor and Taskar [2007]) that combine aspects of first-order logic and probability, a long-standing goal of AI. Instantiating all ground atoms from the formulae in such a model induces a standard graphical model with symmetric, repeated potential structures for all grounding combinations.

To exploit these symmetries, lifted BP (LBP) approaches have been recently developed [Singla and Domingos, 2008; Kersting *et al.*, 2009] to automatically group nodes and potentials of the graphical model into supernodes and superpotentials if they have identical computation trees (i.e., the tree-structured unrolling of the graphical model computations rooted at the nodes). LBP then runs a modified BP on this lifted (clustered) network. LBP has proved extremely fast at computing approximate marginal probability distributions and has yielded significant efficiency gains on important AI tasks such as link prediction, social network analysis, satisfiability and boolean model counting problems.

This paper makes a number of important and novel contributions to both the LBP and GaBP literature while demonstrating the powerful application of these techniques to important AI and novel lifted inference tasks: personalized PageRanks and Kalman filters. To start, we note that the core matrix inversion computation in both PageRank and Kalman filtering can be naïvely solved by combining Shental *et al.*'s [2008] GaBP approach to solving linear systems with Kersting *et al.*'s [2009] color passing approach for LBP, yielding Lifted GaBP (LGaBP). In fact, this novel LGaBP approach — as we will show — already results in considerable efficiency gains. However, we can do considerably better.

Essentially, the computations in personalized PageRanks and Kalman filters require the LGaBP solution of several linear systems with only small changes to the evidence. In turn repeatedly constructing the lifted network for each new inference can be extremely wasteful, because the evidence changes little from one inference to the next. Hence, a less naïve solution would exploit recent efficient approaches for updating the structure of an existing lifted network with small changes to the evidence [Nath and Domingos, 2010; Ahmadi *et al.*, 2010]. In the inference stage, however, this solution would still run a modified LGaBP algorithm on each lifted network separately: symmetries across the inference tasks are not exploited. An investigation of the question of whether we can achieve additional efficiency gains by exploiting symmetries across the inference tasks leads to our main contribution: *multi-evidence lifting*.

In multi-evidence lifting, we first construct the ground networks for all inference tasks. We then run color passing on the union of these networks to compute the joint lifted network that automatically exploits symmetries across inference tasks. Finally, we run a modified message passing algorithm on the joint lifted network that simulates message passing on each ground network in parallel. Intuitively, this sacrifices space complexity for a lower time complexity, and the naïve approach of lifting the joint network will not scale well to large problem sizes. Consequently, we develop an efficient sequential lifting variant that computes the joint lifted network by considering evidence sequentially rather than jointly. As our experiments show, multi-evidence LGaBP[1] and its sequential variant can yield significantly faster inference than naïve LGaBP and GaBP on synthetic problems and the real-world problems of PageRank and Kalman filter computation.

We proceed as follows. After touching upon related work, we briefly review GaBP and color passing for LBP. Then, we introduce multi-evidence lifting, sketch its correctness, and present an efficient sequential lifting variant. Before concluding, we provide empirical results including the first lifted approaches to PageRank and Kalman filter computations.

2 Related Work

Recent years have witnessed a surge of interest in lifted probabilistic inference. Poole [2003], de Salvo Braz *et al.* [2005], and Milch *et al.* [2008] have developed lifted versions of variable elimination for discrete domains; Choi and Amir [2010] have extended these ideas to relational continuous models. These exact inference approaches are extremely complex and have only been applied to artificial domains to date. Moreover, these methods require a first-order logical specification of the model; multi-evidence lifting as presented here does not. Sen *et al.* [2009] presented a lifted (approximate) variable elimination approach based on bisimulation; in contrast to the previous approaches, it does not require a first-order logical specification but is considerably complex. Arguably, the simplest and most efficient approximate lifted inference algorithms are based on the already mentioned LBP approaches. Motivated by Jaimovich *et al.* [2007], Singla and Domingos [2008] developed the first LBP variant requiring a Markov logic network as input. Kersting *et al.* [2009] generalized it to any graphical model over discrete, finite variables.

In contrast to LGaBP introduced here, all previous LBP approaches have been developed for discrete domains only. While in principle they can be applied to continuous domains through discretization, Choi and Amir [2010] have noted the precision of discretizations deteriorates exponentially in the number of random variables. Thus, discretization and application of LBP would be highly imprecise for large networks.

Finally, Nath and Domingos [2010] and Ahmadi *et al.* [2010] developed efficient algorithms for sequential lifting, where the structure of an existing lifted network is updated with incremental evidence changes. However, for inference, these algorithms run message passing independently on the lifted networks constructed for each evidence case. In contrast, the sequential version of multi-evidence lifting we present shares computation via the *joint* lifted network.

3 Lifted Gaussian Belief Propagation

In this section, we present a unified review of Gaussian belief propagation (GaBP) and lifted belief propagation (LBP) leading to lifted GaBP (LGaBP). We develop LGaBP in the context of solving linear systems that are key to our lifted PageRank and Kalman filtering applications presented later.

Many real world applications such as environmental sensor networks, information diffusion in social networks, and localization in robotics involve systems of continuous variables. One of the most fundamental problems encountered in these applications is solving linear systems of the form $\mathbf{A}\mathbf{x} = \mathbf{b}$ where $\mathbf{A} \in \mathbb{R}^{n \times n}$ is a real-valued square matrix, and $\mathbf{b} \in \mathbb{R}^n$ is real-valued column vector, and we seek the column vector $\mathbf{x}$ such that equality holds. As a running example, consider $\mathbf{b} = (0\ 0\ 1)^t$ (where ^{t} denotes transpose) and

$$\mathbf{A} = \begin{pmatrix} 10 & 4 & 3 \\ 4 & 10 & 3 \\ 5 & 5 & 11 \end{pmatrix} . \tag{1}$$

Shental *et al.* [2008] have shown how to translate this problem into a probabilistic inference problem, i.e., to solve a linear system of equations of size n we compute the marginals of the Gaussian variables $x_1, \ldots, x_n$ in an appropriately defined graphical model. Given the matrix $\mathbf{A}$ and the observation matrix $\mathbf{b}$, the Gaussian density function $p(\mathbf{x}) \sim \exp(-\frac{1}{2}\mathbf{x}^t\mathbf{A}\mathbf{x} + \mathbf{b}^t\mathbf{x})$ can be factorized according to the graph consisting of edge potentials ψ_{ij} and self potentials ϕ_i as follows: $p(\mathbf{x}) \propto \prod_{i=1}^{n} \phi_i(x_i) \prod_{i,j} \psi_{ij}(x_i, x_j)$, where the potentials are $\psi_{ij}(x_i, x_j) := \exp(-\frac{1}{2}x_i A_{ij} x_j)$ and $\phi_i(x_i) := \exp(-\frac{1}{2}A_{ii}x_i^2 + b_i x_i)$. The edge potentials ψ_{ij} are defined for all (i, j) s.t. $\mathbf{A}_{ij} > 0$.

To solve the inference task, Shental *et al.* proposed to use Weiss *et al.*'s [2001] Gaussian BP (GaBP) which is a special case of continuous BP, where the underlying distribution is Gaussian. BP in Gaussian models gives simpler update formulas than the general continuous case and the message updates can directly be written in terms of the mean and precision. Since $p(\mathbf{x})$ is jointly Gaussian, the messages are proportional to Gaussian distributions $\mathcal{N}(\mu_{ij}, P_{ij}^{-1})$ with precision $P_{ij} = -A_{ij}^2 P_{i\backslash j}^{-1}$ and mean $\mu_{ij} = -P_{ij}^{-1} A_{ij} \mu_{i\backslash j}$ where

$$P_{i\backslash j} = \tilde{P}_{ii} + \sum\nolimits_{k \in N(i)\backslash j} P_{ki}$$

$$\mu_{i\backslash j} = P_{i\backslash j}^{-1} \big(\tilde{P}_{ii}\tilde{\mu}_{ii} + \sum\nolimits_{k \in N(i)\backslash j} P_{ki}\mu_{ki}\big)$$

for $i \neq j$ and $\tilde{P}_{ii} = A_{ii}$ and $\tilde{\mu}_{ii} = b_i / A_{ii}$. Here, $N(i)$ denotes the set of all the nodes neighboring the ith node and $N(i)\backslash j$ excludes the node j from $N(i)$. All messages parameters P_{ij} and μ_{ij} are initially set to zero. The marginals are Gaussian probability density functions $\mathcal{N}(\mu_i, P_i^{-1})$ with precision $P_i = \tilde{P}_{ii} + \sum_{k \in N(i)} P_{ki}$ and mean $\mu_i = P_{i\backslash j}^{-1}\big(\tilde{P}_{ii}\tilde{\mu}_{ii} + \sum_{k \in N(i)} P_{ki}\mu_{ki}\big)$. If the spectral radius of the matrix $\mathbf{A}$ is smaller than 1 then GaBP converges to the true marginal means ($\mathbf{x} = \mu$). We refer to [Shental *et al.*, 2008] for details.

[1] We focus here on matrix inversion, but multi-evidence lifting is generally applicable, also to discrete domains.

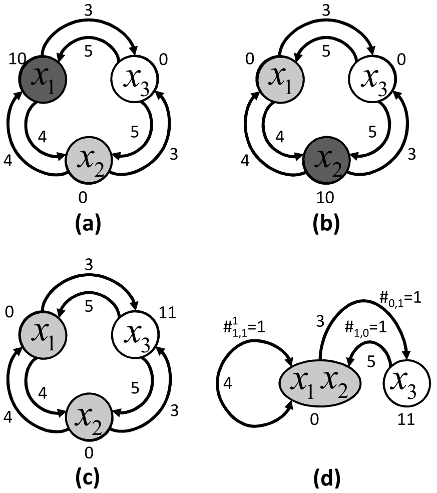

Figure 1: Lifted graphical models produced when inverting $\mathbf{A}$ in Eq. (1) using LGaBP. An edge from i to j encodes potential ψ_{ij}. The ϕ potentials are associated with the nodes. **(a)** Colored network when computing $\mathbf{A}\mathbf{x}_1 = \mathbf{e}_1$. All nodes get different colors; no compression and lifted inference is essentially ground. **(b)** Colored network for $\mathbf{A}\mathbf{x}_2 = \mathbf{e}_2$. Again no compression. Note, however, the symmetries between (a) and (b). **(c)** Colored network for $\mathbf{A}\mathbf{x}_3 = \mathbf{e}_3$. Nodes x_1 and x_2 get the same color and are grouped together **(d)**.

Although already quite efficient, many graphical models produce inference problems with symmetries not reflected in the graphical structure. LBP can exploit this structure by automatically grouping nodes (potentials) of the graphical model G into supernodes (superpotentials) if they have identical *computation trees* (i.e., the tree-structured unrolling of the graphical model computations rooted at the nodes). This compressed graph $\mathcal{G}$ is computed by passing around color signatures in the graph that encode the message history of each node. The signatures are initialized with the color of the self potentials, i.e., $cs_i^0 = \phi_i$ and iteratively updated by $cs_i^k = \{cs_i^{k-1}\} \cup \{[\psi_{ij}, cs(\psi_j^{k-1})] | \, j \in N(i)\}$. The algorithmic details of color passing are not important for this paper, we refer to [Kersting *et al.*, 2009]. The key point to observe is that this very same process also applies to GaBP (viewing "identical" for potentials only up to a finite precision), thus leading to a novel LGaBP algorithm.

To continue our running example, let us examine computing the inverse of matrix $\mathbf{A}$ in Eq. (1) using LGaBP. We note that $\mathbf{A}^{-1} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ can be computed by solving $\mathbf{A}\mathbf{x}_i = \mathbf{e}_i$ for $i = 1 \ldots n$, where $\mathbf{e}_i$ is the ith *basis vector* — $\mathbf{I} = [\mathbf{e}_1, \ldots, \mathbf{e}_n]$ for the $n \times n$ identity matrix $\mathbf{I}$. In our running example, this yields the respective lifted networks in Figs. 1(a–c). As one can see, for the evidence cases e_1 and e_2 there is no compression and lifted inference is essentially ground. All nodes get different colors for these cases (Figs. 1(a) and (b)). For the evidence case $\mathbf{e}_3$, however, variables x_1 and x_2 are assigned the same color by LGaBP (Fig. 1(c)). The final lifted graph $\mathcal{G}$ is constructed by grouping all nodes (potentials) with the same color (signatures) into *supernodes* (*superpotentials*), which are sets of nodes (potentials) that behave identical at each step of carrying out GaBP on G (Fig. 1(d)). On the lifted graph $\mathcal{G}$, LGaBP then runs a modified GaBP. The modified messages simulate running GaBP on the original graph G. Following LBP, we have to pay special attention to the self-loops introduced by lifting that correspond to messages between different nodes of the same supernode. Reconsider our running example. As shown in Fig. 1(d), there is a self-loop for the supernode $\{x_1, x_2\}$. In general, there might be several of them for each supernode and we assume that they are indexed by k. To account for the self-loops and in contrast to GaBP, we introduce "self-messages" $P_{ii}^k = -A_{ii}^2 (P_{i\backslash i}^k)^{-1}$ and $\mu_{ii}^k = -P_{ij}^{-1} A_{ii} \mu_{i\backslash i}^k$ with

$$P_{i\backslash i}^k = \tilde{P}_{ii} + \Big(\sum_{l \in S(i)\backslash k} \sharp_{ii}^l P_{ii}^l \Big) + \Big(\sum_{l \in N(i)\backslash i} \sharp_{li} P_{li} \Big)$$

$$\mu_{i\backslash i}^k = P_{i\backslash j}^{-1} \Big[\tilde{P}_{ii} \tilde{\mu}_{ii} + \sum_{l \in S(i)\backslash k} \sharp_{ii}^l P_{ii}^l \mu_{ii}^l + \sum_{l \in N(i)\backslash i} \sharp_{li} P_{li} \mu_{li} \Big] \,.$$

As i is now a neighbor of itself, the term $N(i) \setminus i$ is required. Furthermore, $\sharp_{ij}^l$ — also given in Fig. 1(d) — are counts that encode how often the message (potential) would have been used by GaBP on the original network G. Using these counts we can exactly simulate the messages that would have been sent in the ground network. Messages between supernode i and j, $i \neq j$, are modified correspondingly:

$$P_{i\backslash j} = \tilde{P}_{ii} + \sum_{k \in S(i)} \sharp_{ii}^k P_{ii}^k + \Big(\sum_{k \in N(i)\backslash i,j} \sharp_{ki} P_{ki} \Big)$$

$$\mu_{i\backslash j} = P_{i\backslash j}^{-1} \Big[\tilde{P}_{ii} \tilde{\mu}_{ii} + \sum_{k \in S(i)} \sharp_{ii}^k P_{ii}^k \mu_{ii}^k + \sum_{k \in N(i)\backslash i,j} \sharp_{ki} P_{ki} \mu_{ki} \Big]$$

where $N(i) \setminus i, j$ denotes all neighbours of node i without i and j. Adapting the arguments from [Kersting *et al.*, 2009], the following LGaBP correctness theorem can be proved:

Theorem 3.1. *Given a Gaussian model G, LGaBP computes the minimal compressed lifted model, and running modified GaBP on $\mathcal{G}$ produces the same marginals as GaBP on G.*

4 Multi-Evidence Lifting

Both applications considered in this paper — lifted PageRank and Kalman filtering — require the inversion of very large, structured matrices. As shown previously, this task can be reduced to the problem of solving several linear systems with GaBP — more precisely, calling GaBP multiple times, each time with a different evidence case.

Returning to our running example, inverting $\mathbf{A}$ from Eq. (1) using GaBP results in three graphical models of size $3/9$ (nodes/potentials). Given the recent success of LBP approaches, we ask "can we do better?" A first attempt to affirmatively answer the question is to simply replace GaBP with LGaBP for each evidence case $\mathbf{e_i}$ as illustrated in the three lifted graphical models of Figs. 1(a–d). In general, this *single-evidence lifting* approach is illustrated in Fig. 2(a). Due to lifting, we can hope to greatly reduce the cost of inference in each iteration. For our running example, this results in two ground networks both of size $3/9$ and one lifted network of size $2/5$ shown respectively in Figs. 1(a,b) and (d). Thus, the total size (sum of individual sizes) drops to $8/23$ This LGaBP approach — as we will show in our experiments

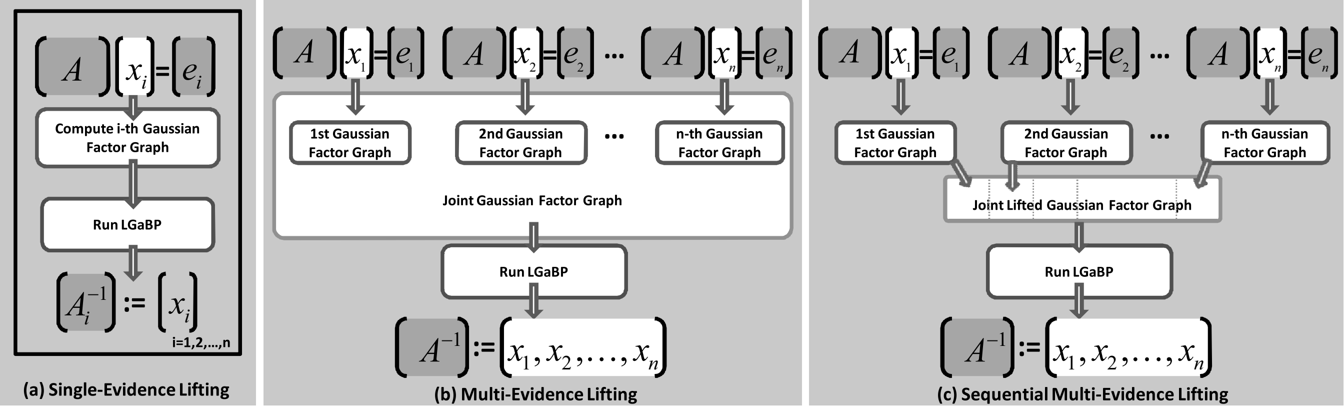
A x_i = e_i
Compute i-th Gaussian Factor Graph
Run LGaBP
A_i^{-1} := x_i
i=1,2,…,n
(a) Single-Evidence Lifting
A x_1 = e_1 A x_2 = e_2 ··· A x_n = e_n
1st Gaussian Factor Graph
2nd Gaussian Factor Graph
•••
n-th Gaussian Factor Graph
Joint Gaussian Factor Graph
Run LGaBP
A^{-1} := x_1, x_2, …, x_n
(b) Multi-Evidence Lifting
A x_1 = e_1 A x_2 = e_2 ··· A x_n = e_n
1st Gaussian Factor Graph
2nd Gaussian Factor Graph
•••
n-th Gaussian Factor Graph
Joint Lifted Gaussian Factor Graph
Run LGaBP
A^{-1} := x_1, x_2, …, x_n
(c) Sequential Multi-Evidence Lifting

Figure 2: Inverting a matrix $\mathbf{A}$ using multi-evidence lifting. (a) Single-evidence lifting runs LGaBP solving $\mathbf{A}\mathbf{x}_i = \mathbf{e}_i$ for each $i = 1, 2, \ldots, n$ separately; $\mathbf{e}_i$ denotes the ith basis vector; thus LGaBP computes each column vector of $\mathbf{A}^{-1}$ separately. (b) Multi-evidence lifting runs LGaBP on the joint graphical model of $\mathbf{A}\mathbf{x}_i = \mathbf{e}_i$ for $i = 1, 2, \ldots, n$; thus LGaBP computes $\mathbf{A}^{-1}$ for all e_i at once. (c) Sequential multi-evidence lifting directly builds the lifted joint graphical model in a sequential fashion, avoiding cubic space complexity

— can already result in considerable efficiency gains. However, we can do even better.

Repeatedly constructing the lifted network for each new evidence case can be wasteful when symmetries across multiple evidence cases are not exploited (e.g., the common 0's amongst all $\mathbf{e}_i$). Compare for example the graphs depicted in Fig. 1(a) and (b). These two networks are basically symmetric which is not exploited in the single-evidence case.They stay essentially ground if they are processed seperately. To overcome this, we propose *multi-evidence lifting* (as illustrated in Fig. 2(b) for inverting a matrix):

> *compute the graphical models of each evidence case, form their union, and run LGaBP on the resulting joint graphical model.*

This automatically employs the symmetries within and across the evidence cases due to Theorem 3.1. In our running example, we get a *single* lifted graph of size $5/14$. Intuitively, multi-evidence lifting only produces (in this example) one of the two ground networks and hence consists of the union of the networks shown in Figs. 1(a) and (d). This is clearly a reduction compared to LGaBP's size of $8/23$.

However, there is no free lunch. Multi-evidence lifting sacrifices space complexity for a lower time complexity. Inverting a $n \times n$ matrix may result in a joint ground graphical model with n^2 nodes (n nodes for each of the n systems of linear equations) and $O(n^3)$ edges ($O(n^2)$ edges for each of the n systems of linear equations; edges are omitted if $\mathbf{A}_{ij} = 0$). This makes multi-evidence essentially intractable for large n. For instance, already for $n > 100$ we have to deal with millions of edges, easily canceling the benefits of lifted inference. We develop an efficient sequential multi-evidence lifting approach that computes the joint lifted network by considering one evidence case after the other (Fig. 2(c)).

Indeed, one is tempted to employ one of the efficient algorithms recently presented for updating the structure of an existing lifted network with incremental changes to the evidence to solve the problem [Nath and Domingos, 2010; Ahmadi *et al.*, 2010]. While employing the symmetries in the graphical model across multiple evidence cases for the lifting, in the inference stage they need to construct a separate lifted network for each evidence case and run a modified message passing algorithm on each lifted network separately. Thus, symmetries across evidence cases are missed.

5 Sequential Multi-Evidence Lifting

We seek a way to efficiently construct the joint lifted network while still being able to lift across multiple evidence cases. To do this, we can modify Ahmadi *et al.*'s [2010] sequential single-evidence lifting to the multi-evidence case. Ahmadi *et al.* give a clear characterization of the core information required for sequential clamping for lifted message passing, namely the shortest-paths connecting the variables in the network which resemble the computation paths along which the nodes communicate in the network.

Thus, to be able to adapt the lifted network for incoming evidence in the single-evidence case, we compute in a first step the set of shortest paths connecting any two nodes in the graph. Now, when there is new evidence for a node, the adapted lifted network is computed as a combination of the nodes' initial coloring, i.e. the lifting without evidence, and the set of shortest paths to the nodes in our evidence.

Algorithm 1 describes how we can adapt this to the multi-evidence case. Intuitively, we would like to view each $\mathbf{A}\mathbf{x}_i = \mathbf{e}_i$ ($i = 1, 2, \ldots, n$) as conditioning an initial network on node i and efficiently computing its contribution to the resulting lifted joint network directly from the initial one. But what should be the initial network? It is not provided by the task itself. Thus, to be able to efficiently find the lifted network structure, we propose to introduce an additional system of linear equations, namely the one with no evidence: $\mathbf{A}\mathbf{x} = \mathbf{0}$ (Alg.1, line 1). Indeed, it is not needed in the inversion task per se but it allows us to significantly speed up the multi-evidence lifting process. Intuitively, each $\mathbf{e}_i$ only differs from $\mathbf{0}$ in exactly one element so it serves as the basis for lifting all subsequent networks. To do this, we compute the path colors for all pairs of nodes (line 2) to know how it affects the other nodes when conditioning on a variable i and the initial lifting without evidence (line 3). The lifted graph H_i

Algorithm 1: Sequential Multi-Evidence Lifted GaBP

```
   Input: Matrix $\mathbf{A}$
1  Construct network $G_0$ for $\mathbf{Ax} = \mathbf{0}$;
2  Compute path color matrix $\mathbf{PC}$ on $G_0$;
3  Lift $G_0$ to obtain lifted network $H_0$;
4  foreach unit vector $e_i$ do
5      colors($H_i$) = colors($H_0$);
6      evidence = $\{i{:}A_{ii}\}$;
7      repeat
8          colors($H_i$) = newColor ($H_0$, $PC$, evidence);
9          Add nodes that have changed color to evidence;
10     until colors($H_i$) does not change ;
11     if $H_i$ is previously unseen then
12         Add $H_i$ to joint lifted network $H_{\{1,..,i-1\}}$
13     else
14         Bookmark the corresponding index $j$
15
16 Run modified GaBP on joint lifted network $H_{\{1,..,n\}}$;
17 return $\mathbf{X} = (x_1, x_2, \ldots, x_n)$, the inverse of $\mathbf{A}$;
```

for the ith system of linear equations $\mathbf{Ax}_i = \mathbf{e}_i$ can now be adaptively computed by combining the initial lifting and the path colors to node i (line 8). The combination is essentially an elementwise concatenation of the two respective vectors.

Consider computing the lifted network for $\mathbf{Ax_3} = \mathbf{e_3}$ for our running example. When we have no evidence the nodes x_1 and x_2 are clustered together, and x_3 is in a separate cluster. Thus, we obtain an initial color vector $C = (0, 0, 1)$. Since we want to compute the network conditioned on x_3 we have to combine this initial clustering C with the path colors with respect to x_3, $PC_{x_3} = (\{3, 5\}, \{3, 5\}, \emptyset) = (0, 0, 1)$. To obtain the lifted network conditioned on x_3 we have to (1) do an elementwise concatenation (in the following depicted by $\oplus$) of the two vectors and (2) interpret the result as a new color vector: $H_3 = (0, 0, 1) \oplus (0, 0, 1) =_{(1)} (00, 00, 11) =_{(2)} (2, 2, 3)$. Since only the shortest paths are computed, adapting the colors has to be performed iteratively to let the evidence propagate. For further details on adapting the color vector (line 8) we refer to [Ahmadi *et al.*, 2010].

Moreover, we can implement a type of memoization when we perform the lifting in this fashion. Because we know each resulting lifted network H_i in advance, we can check whether an equivalent lifted network was already constructed: if the same color pattern exists already, we simply do not add H_i and instead only bookmark the correspondence of nodes (line 11-14) [2]. This does not affect the counts at all and, hence, still constructs the correct joint lifted network. This argument together with the correctness of the sequential single-evidence lifting and multi-evidence lifting effectively proves the correctness of sequential multi-evidence lifting:

Theorem 5.1. *Sequential multi-evidence lifting computes the same joint lifted model as in the batch case. Hence, running the modified GaBP on it produces the same marginals.*

6 Experimental Evaluation

Our intention here is to investigate the following questions: **Q1** Can LGaBP be faster than GaBP? **Q2** Can multi-evidence lifting produce smaller inference problems than single-evidence lifting? **Q3** Does sequential multi-evidence lifting scale better than just multi-evidence lifting.

We implemented all variants in Python using the LIBDAI library[3] and evaluated their performances on (a) random matrices, (b) PageRank computations of graphs induced by a Markov logic network (MLN), and (c) Kalman filtering problems. The GaBP variants ran using parallel message updates, no damping and convergence threshold $\epsilon = 10^{-8}$. They all converged within ϵ of the correct solution.

Inverting Random Matrices: We generated a random matrix $\mathbf{R} \in \mathbb{R}^{20 \times 20}$ with $\mathbf{R}_{ij} \in [0, 1]$ and added 10 to the diagonal to ensure non-singularity. Using $\mathbf{R}$, we constructed a diagonal matrix with $1, 2, 4$ and 8 blocks. On each of the matrices we run all four algorithms measuring the number of potentials created and total messages sent (including coloring messages), and the CPU time. Figs. 3(a)-(c) show the results averaged over 10 random reruns. As one can see, SME-LGaBP $<$ ME-LGaBP $<$ LGaBP $<$ GaBP in terms of the number of messages sent (b) and CPU-time (sec.) (c). Furthermore, (sequential) ME-LGaBP and create — as expected — the same number of potentials and significantly less than (L)GaBP (b).

Lifted Personalized PageRank: Recall the problem of ranking web pages already touched upon in the introduction. In a nutshell, a Markov chain transition matrix $\mathbf{M}$ is constructed out of a given graph G. Let $\mathbf{A}$ be the $n \times n$ adjanceny matrix of G, that is $\mathbf{A}_{ij} = 1$ if there is an edge from vertex j to vertex i and zero otherwise. PageRank now constructs the probability transition matrix $\mathbf{M}$ by adding $1/n$ to all entries of $\mathbf{A}$ and renormalizes each row of A to sum to 1. Furthermore, a prior probability $\mathbf{v}$ can be taken to weight the results. The personalized PageRank can then be computed by solving the following system of linear equations $(\mathbf{I} - \alpha\mathbf{M})\mathbf{x} = \mathbf{v}$ where α trades off speed of convergence with the accuracy of the solution and $\mathbf{I}$ is the identity matrix.

In several PageRank applications, however, one needs to know, in addition to the rank of a given page, which pages or sets of pages contribute most to its rank. These PageRank contributions have been used for link spam detection and in the classification of web pages. The contribution that a vertex v makes to the PageRank of a vertex u is defined rigorously in terms of personalized PageRank for all vertices, i.e., for $\mathbf{v}_i = \mathbf{e}_i$, $i = 1, 2, \ldots, n$. So, we are interested in $\mathbf{PRM}_\alpha$ — the matrix whose u-th row is the personalized PageRank vector of u. The PageRank contribution of u to v is then the entry (u, v) of this matrix. This, however, is the inverse of $(\mathbf{I} - \alpha\mathbf{M})$ and we can make use of multi-evidence lifting.

We computed $\mathbf{PRM}_{0.9}$ for the ground network induced by the *Friends & Smokers* Markov logic network [Richardson and Domingos, 2006]. We varied the number of people in the domain, namely $3, 5, 10$. Figs. 3 (d)–(f) show the (d) the total number of (lifted) potentials (e) total numbers of messages

[2] This is not as hard as solving (sub)graph-isomorphisms. We only have to check whether the color pattern of H_i was previously seen. If so, the result has already been memoized.

[3] http://www.libdai.org/

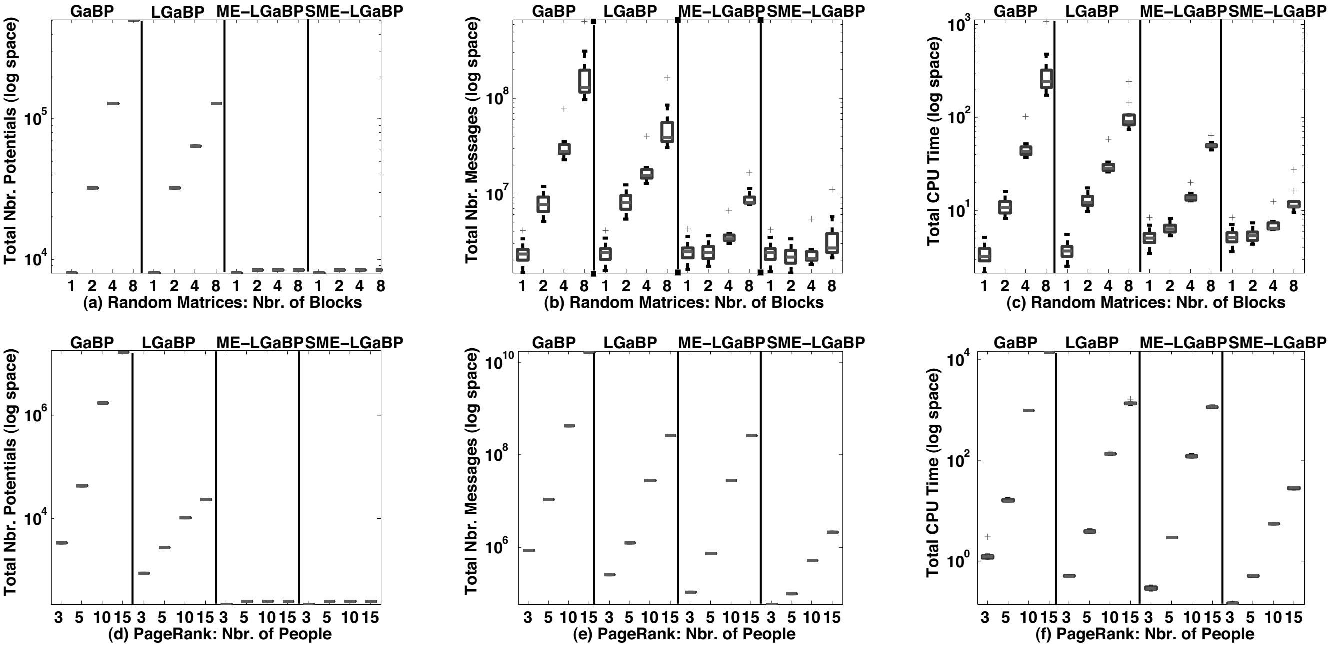

Figure 3: Experimental results on random matrices and PageRank computations. (best viewed in color)

sent and (f) the total CPU times (sec.) per $\mathbf{PRM}_{0.9}$ computation averaged over 10 runs. Again one can see, SME-LGaBP < ME-LGaBP < LGaBP < GaBP. In particular, we see the benefit of multi-evidence lifting. The number of generated potentials saturates for more than 3 people whereas single-evidence lifting generates more and more potentials.

Lifted Kalman Filter: Kalman Filtering is a computational tool with widespread application in robotics, financial and weather forecasting, and environmental engineering. Given observation and state transition models, the Kalman Filter (KF) recursively estimates the state $\mathbf{x} \in \mathbb{R}^n$ of a discrete-time controlled process that is governed by the linear stochastic difference equation $\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \mathbf{w}_{k-1}$ with a measurement $\mathbf{z} \in \mathbb{R}^m$ that is $\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k$. Here, $\mathbf{w}_k$ and $\mathbf{v}_k$ represent the process and measurement noise (respectively) and are assumed to be independent (of each other), white, and with normal density, i.e., $p(\mathbf{w}) \sim \mathcal{N}(0, \mathbf{Q})$ and $p(\mathbf{v}) \sim \mathcal{N}(0, \mathbf{R})$. The matrix $\mathbf{A}$ relates consecutive states $\mathbf{x}_{k-1}$ and $\mathbf{x}_k$. The matrix $\mathbf{B}$ relates the optional control input $\mathbf{u}$ to the state, and $\mathbf{H}$ relates the state to the measurement $\mathbf{z}$. In practice, the matrices $\mathbf{A}, \mathbf{B}, \mathbf{H}, \mathbf{Q}$ and $\mathbf{R}$ might change with each time step or measurement, however, here we assume they are constant.

Due to space limitations, we cannot go into details but instead refer to, e.g, [Thrun *et al.*, 2005], and note that the main step of the KF consists of computing the Kalman gain: $\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^t \big(\mathbf{H}\mathbf{P}_k^- \mathbf{H}^t + \mathbf{R}\big)^{-1}$ where $\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^t + \mathbf{Q}$. Thus, the KF requires to invert a matrix at every time step allowing us to apply multi-evidence lifting.

In our experiments, we tracked 10 people randomly spread among k groups. Each group had its own (local) motion model (mm). We varied the number of groups: 1 (all people have the same mm) , 5 (two people have the same mm), and 10 (everybody has its own mm). Figs. 4 shows (a) the total number of (lifted) potentials, (b) the total number of messages sent and (c) the CPU times (sec.) averaged over all matrix inversion tasks in a Kalman filtering over 10 steps (GaBP is omitted). It clearly shows: the larger the number of groups, the lower the gain of (sequential) multi-evidence lifting. For 10 groups, when there are no mm symmetries — motion symmetries ratio 0 — across people, single-evidence lifting is faster. However, when we have symmetries across evidence cases — 5 and 1 groups, i.e., ratios of 0.5 and 1.0 — ME lifting significantly outperforms single-evidence lifting.

The Discrete Case: Indeed, we introduced multi-evidence lifting in the context of GaBP. It is, however, also applicable in discrete cases. To validate its generality, we peformed parameter estimation – another natural case for multi-evidence lifting – in discrete domains. For the *Friends & Smokers* MLN with ten people. we maximized the conditional marginal log-likelihood (CMLL) using scaled conjugate gradient (SCG) for 10 data cases sampled from the joint distribution. Here, lifted BP (LBP) took 0.089 seconds for a single iteration — already a reduction compared to BP's 0.1 sec. — but ME-LBP exploits the additional symmetries and in turn took only 0.049 sec.

All experimental results together clearly answer questions **Q1**-**Q3** affirmatively.

7 Conclusions

In this paper, we proposed multi-evidence lifting for belief propagation in graphical models that exploits symmetries within and across different evidence cases. To avoid the cubic space requirement of a naïve realization, we then presented a sequential algorithm for efficiently computing the multi-evidence lifted network. The experimental results on two novel tasks for lifted inference, namely computing PageRank contributions and Kalman filter updates show that

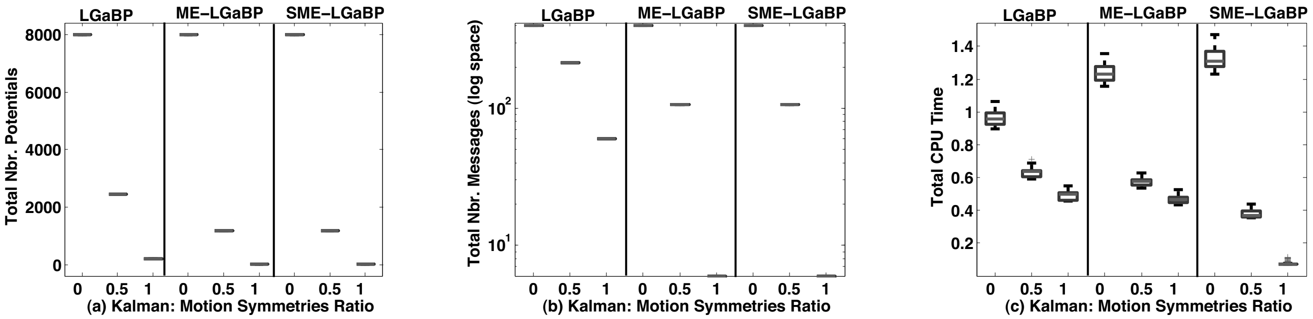

Figure 4: Experimental results for Kalman filtering with varying degree of motion symmetries. (best viewed in color)

multi-evidence lifting provides substantial speedups — up to two orders of magnitude — over standard lifting, making it applicable to a wide range of problems.

Indeed, much remains to be done. Since lifting of GaBP itself and the exploitation of multi-evidence sequential lifting for solving large linear systems is a major advance, further experimentation and application of these techniques to other tasks requiring the solution of linear systems and matrix inversion is one important direction. In addition, exploring its usefulness in other AI and machine learning tasks is another interesting avenue for future work; promising candidates are lifted variants of ridge regression, least-squares support vector machines, and linear program solvers.

Acknowledgments: The authors thank the anonymous reviewers for their comments. BA and KK were supported by the Fraunhofer ATTRACT fellowship STREAM and by the European Commission under contract number FP7-248258-First-MM. NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

References

[Ahmadi *et al.*, 2010] B. Ahmadi, K. Kersting, and F. Hadiji. Lifted belief propagation: Pairwise marginals and beyond. In *Proceedings of the 5th European WS on Probabilistic Graphical Models (PGM–10)*, Helsinki, Finland, 2010.

[Bickson *et al.*, 2007] D. Bickson, D. Malkhi, and L. Zhou. Peer to peer rating. In *the 7th IEEE Peer-to-Peer Computing*, pages 211–218, Galway, Ireland, Sept. 2007.

[Brin and Page, 2006] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30:197–117, 2006.

[Choi and Amir, 2010] J. Choi and E. Amir. Lifted inference for relational continuous models. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI-10)*, Catalina Island, USA, July 8 – 10 2010.

[de Salvo Braz *et al.*, 2005] R. de Salvo Braz, E. Amir, and D. Roth. Lifted First Order Probabilistic Inference. In *Proc. of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05)*, pages 1319–1325, 2005.

[Getoor and Taskar, 2007] L. Getoor and B. Taskar, editors. *An Introduction to Statistical Relational Learning*. MIT Press, 2007.

[Jaimovich *et al.*, 2007] A. Jaimovich, O. Meshi, and N. Friedman. Template-based Inference in Symmetric Relational Markov Random Fields. In *Proc. of the Conf. on Uncertainty in AI (UAI-07)*, pages 191–199, 2007.

[Kersting *et al.*, 2009] K. Kersting, B. Ahmadi, and S. Natarajan. Counting Belief Propagation. In *Proceedings of the 25th Conference on Uncertainty in AI (UAI–09)*, Montreal, Canada, June 18–21 2009.

[Milch *et al.*, 2008] B. Milch, L. Zettlemoyer, K. Kersting, M. Haimes, and L. Pack Kaelbling. Lifted Probabilistic Inference with Counting Formulas. In *Proc. of the 23rd AAAI Conf. on AI (AAAI-08)*, July 13-17 2008.

[Nath and Domingos, 2010] A. Nath and P. Domingos. Efficient lifting for online probabilistic inference. In *Proceedings of the 24th AAAI Conf. on AI (AAAI-10)*, 2010.

[Pearl, 1991] J. Pearl. *Reasoning in Intelligent Systems: Networks of Plausible Inference*. MK, 1991.

[Poole, 2003] D. Poole. First-Order Probabilistic Inference. In *Proc. of the 18th International Joint Conference on Artificial Intelligence (IJCAI-05)*, pages 985–991, 2003.

[Richardson and Domingos, 2006] M. Richardson and P. Domingos. Markov Logic Networks. *Machine Learning*, 62:107–136, 2006.

[Sen *et al.*, 2009] P. Sen, A. Deshpande, and L. Getoor. Bisimulation-based Approximate Lifted Inference. In *Proc. of the 25th Conf. on Uncertainty in AI (UAI–09)*, Montreal, Canada, 2009.

[Shental *et al.*, 2008] O. Shental, D. Bickson, P. H. Siegel, J. K. Wolf, and D. Dolev. Gaussian belief propagation solver for systems of linear equations. In *IEEE Int. Symp. on Inform. Theory (ISIT)*, Toronto, Canada, July 2008.

[Singla and Domingos, 2008] P. Singla and P. Domingos. Lifted First-Order Belief Propagation. In *Proc. of the 23rd AAAI Conf. on Artificial Intelligence (AAAI-08)*, pages 1094–1099, Chicago, IL, USA, July 13-17 2008.

[Thrun *et al.*, 2005] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT Press, 2005.

[Weiss and Freeman, 2001] Y. Weiss and W.T. Freeman. Correctness of belief propagation in gaussian graphical models of arbitrary topology. *Neural Computation*, 13(10):2173–330, 2001.