

Lifted Relational Kalman Filtering

Jaesik Choi and Abner Guzman-Rivera and Eyal Amir

Computer Science Department

University of Illinois at Urbana-Champaign

Urbana, IL, 61801, USA

{jaesik,aguzman5,eyal}@illinois.edu

Abstract

Kalman Filtering is a computational tool with widespread applications in robotics, financial and weather forecasting, environmental engineering and defense. Given observation and state transition models, the Kalman Filter (KF) recursively estimates the state variables of a dynamic system. However, the KF requires a cubic time matrix inversion operation at every timestep which prevents its application in domains with large numbers of state variables. We propose Relational Gaussian Models to represent and model dynamic systems with large numbers of variables efficiently. Furthermore, we devise an exact lifted Kalman Filtering algorithm which takes only linear time in the number of random variables at every timestep. We prove that our algorithm takes linear time in the number of state variables even when individual observations apply to each variable. To our knowledge, this is the first lifted (linear time) algorithm for filtering with continuous dynamic relational models.

1 Introduction

Many real-world systems can be modeled by continuous variables and relationships (or dependences) among them. The Kalman Filter (KF) [Kalman, 1960] accurately estimates the state of a dynamic system given a sequence of control-inputs and observations. It has been applied in a broad range of domains which include weather forecasting [Burgers *et al.*, 1998], localization and tracking in robotics [Limketkai *et al.*, 2005], economic forecasting in finance [Bahmani-Oskooee and Brown, 2004] and many others. Given a sequence of observations and Gaussian dependences between variables, the filtering problem is to calculate the conditional probability density of the state variables at each timestep. Unfortunately, the KF computations are cubic in the number of random variables which limits current exact methods to domains with limited number of random variables. This has led to the combination of approximation and sampling (e.g. the Ensemble Kalman Filter [Evensen, 1994]).

This paper leverages the ability of relational languages [Friedman *et al.*, 1999; Poole, 2003; Richardson and Domingos, 2006] to specify models with size of representation

independent of the size of populations involved. Various lifted inference algorithms for relational models have been proposed [Poole, 2003; de Salvo Braz *et al.*, 2005; Milch and Russell, 2006; Richardson and Domingos, 2006; Wang and Domingos, 2008; Choi *et al.*, 2010]. These seek to carry computations in time independent of the size of the populations involved. However, the key challenge in relational filtering (of dynamic systems) is ensuring that the representation does not degenerate to the ground case when multiple observation are made. As more observations are received, an increasing number of objects become distinguished. This precludes the application of previously known algorithms unless approximately equivalent objects are grouped with expensive clustering algorithms.

We propose Relational Gaussian Models (RGMs) to model dynamic systems of large number of variables in a relational fashion. RGMs have as their main building block the pairwise linear Gaussian potential as detailed in Section 2. Further, we propose a new lifted filtering algorithm that is able to marginalize out random variables of the previous timestep efficiently (in time linear in the number of random variables) while maintaining the relational (RGM) representation. This prevents the model from being increasingly grounded even when individual observations are made for all random variables. Moreover, updating the relational representation takes only quadratic time in the number of relational atoms (sets of random variables). One key insight is that, given identical observation models, even when the means of the random variables are dispersed their variances remain identical. This is sufficient to maintain a relational representation.

This paper is organized as follows. Section 2 introduces definitions and the relational filtering problem. Section 3 presents our main technical results, i.e., the recursive estimation of the states of random variables in a lifted fashion. Section 4 presents our algorithm in detail together with complexity results. Section 5 shows experimental results with a housing market model. Section 6 presents a real-world application to Social Networks. Section 7 discusses previous work. We conclude in Section 8.

2 Model and Problem Definitions

In this section, we define Relational Gaussian Models (RGMs) and introduce the filtering problem for dynamic relational models.

2.1 Relational Continuous Models

Dependencies between variables are represented using **Parfactor** models¹, i.e. parameterized factor models. Each *parfactor* $g = (L, C, \mathbf{X}, \phi)$ is composed of a set of logical variables (or objects) (L), constraints on L (C), a list of relational atoms (\mathbf{X}), and a potential function on \mathbf{X} (ϕ).

Relational atoms represent the set of random variables corresponding to all ground substitutions of its logical variables. Formally, applying a substitution θ to a parfactor g yields a new parfactor $g\theta = (L', C\theta, \mathbf{X}\theta, \phi)$, where L' is obtained by renaming the variables in L according to θ . If θ is a ground substitution, $g\theta$ is a factor. A factor $f = (\mathbf{x}, \phi)$ is a pair where \mathbf{x} is a list of ground random variables ($x_1, \dots, x_{|\mathbf{x}|}$) and ϕ is a potential on \mathbf{x} , a function from $\text{range}(\mathbf{x}) = \times_{i=1}^{|\mathbf{x}|} \text{range}(x_i)$ to \mathbb{R}^+ . A factor f defines a weighting function on a valuation $v = (v_1, \dots, v_{|\mathbf{x}|})$: $w_f(v) = \phi(v_1, \dots, v_{|\mathbf{x}|})$. The weighting function for a *parfactor* g is the product of the weighting functions of all of its ground substitutions (factors), $w_g(v) = \prod_{f \in \mathcal{G}} w_f(v)$. Hence, a set of parfactors G defines² a probability density proportional to,

$$w_G(v) = \prod_{g \in G} \prod_{f \in \mathcal{G}} w_f(v). \quad (1)$$

2.2 Relational Gaussian Models (RGMs)

Relational Gaussian Models (RGMs) are a subset of Relational Continuous Models (RCMs) where potentials are restricted to be Gaussian distributions. RGMs are composed of three types of parfactor models: (1) Relational Transition Models (RTMs); (2) Relational Pairwise Models (RPMs); and (3) Relational Observation Models (ROMs). Suppose that we have n relational atoms: $X_t^1(L), \dots, X_t^n(L)$ where L is a list of logical variables. In a relational linear dynamic model, relational atoms are linearly influenced by control-inputs $U_t^1(L), \dots, U_t^n(L)$. Similarly, a linear observation model specifies the relationship between observation variables $O_t^1(L), \dots, O_t^n(L)$ and other relational atoms. Control inputs and observations are associated with relational atoms in two ways: (1) direct association; and (2) indirect association. We provide further details in Section 2.4.

Relational Transition Models (RTMs) model the dependence of relational atoms at the next timestep, $X_{t+1}^j(a')$, on relational atoms at the current timestep, $X_t^i(a)$, and (when available) control-input information. They take the following form,

$$X_{t+1}^j(a') = B_X^{i,j} \cdot X_t^i(a) + B_U^{i,j} \cdot U_t^i(a) + G_{RTM}^{i,j} \quad (2)$$

where $G_{RTM}^{i,j} \sim N(0, \sigma_{RTM}^{i,j})$ and $N(m, \sigma^2)$ is the normal distribution with mean m and variance σ^2 . $B_X^{i,j}$ and $B_U^{i,j}$ are the transition models, matrices or a constants, corresponding to two relational atoms.

¹Our representation is based on previous work [Poole, 2003; de Salvo Braz *et al.*, 2005; Milch and Russell, 2006; Choi *et al.*, 2010].

²The condition is that at least a random variable has a prior distribution as outlined in [Choi *et al.*, 2010].

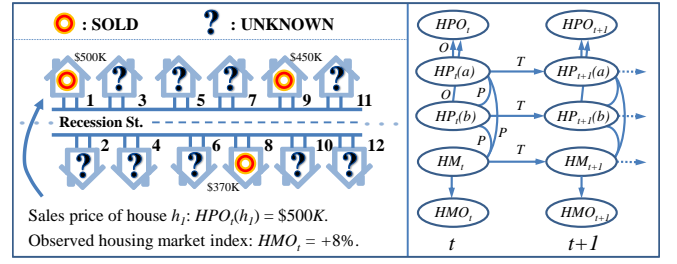


Figure 1: Example of a housing market model. We are interested in estimating the hidden value of houses given observations of house sales prices (e.g. $HPO_t(1) = \$500K$). Both, the hidden value of a house and the observed sales prices are affected by several factors, e.g., house values increase by a certain rate every year and are also influenced by a housing market index (HM_t).

For univariate state variables, we can represent the transition model with a linear Gaussian,

$$\phi_{RTM}(X_{t+1}^j(a') | X_t^i(a), U_t^i(a)) \propto \exp \left(- \frac{(X_{t+1}^j(a') - B_X^{i,j} \cdot X_t^i(a) - B_U^{i,j} \cdot U_t^i(a))^2}{2 \cdot \sigma_{RTM}^{i,j}{}^2} \right). \quad (3)$$

The most common transition is the transition from the current state $X_t^i(a)$ to the next $X_{t+1}^i(a)$. It is represented as follows,

$$X_{t+1}^i(a) = B_X^i \cdot X_t^i(a) + B_U^i \cdot U_t^i(a) + G_{RTM}^i. \quad (4)$$

Relational Observation Models (ROMs) represent the relationships between the hidden (state) variables, $X_t^i(a)$, and the observations made at the corresponding timestep, $O_t^i(a)$,

$$O_t^i(a) = H_t^i \cdot X_t^i(a) + G_{ROM}^i, \quad (5)$$

where $G_{ROM}^i \sim N(0, \sigma_{ROM}^i)$. H_t^i is the observation model, a matrix or a constant, between the hidden variables and the observations.

In the linear Gaussian representation, they take the following form,

$$\phi_{ROM}(O_t^i(a) | X_t^i(a)) \propto \exp \left(- \frac{(O_t^i(a) - H_t^i \cdot X_t^i(a))^2}{2 \cdot \sigma_{ROM}^i{}^2} \right). \quad (6)$$

Relational Pairwise Models (RPMs) represent Gaussian dependences between pairs of relational atoms within the same timestep as follows,

$$X_t^i(a) = R_t^{i,j} \cdot X_t^j(a') + G_{RPM}^{i,j} \quad (7)$$

where $G_{RPM}^{i,j} \sim N(0, \sigma_{RPM}^{i,j})$. $R_t^{i,j}$ is the pairwise coefficient, a matrix or a constant, between the two relational atoms.

Note that RTMs and ROMs are directed models while RPMs are undirected. The directed models represent the nature of dynamic systems (e.g. the state at the next timestep depends on the current timestep). The product of RPMs is an efficient way to represent a multivariate Gaussian density over all the state variables.³

³Note that a multivariate Gaussian density (of state variables) is a quadratic exponential form. The quadratic exponential form can always be decomposed into terms involving only single variables and pairs of variables. We provide additional details in Section 7.

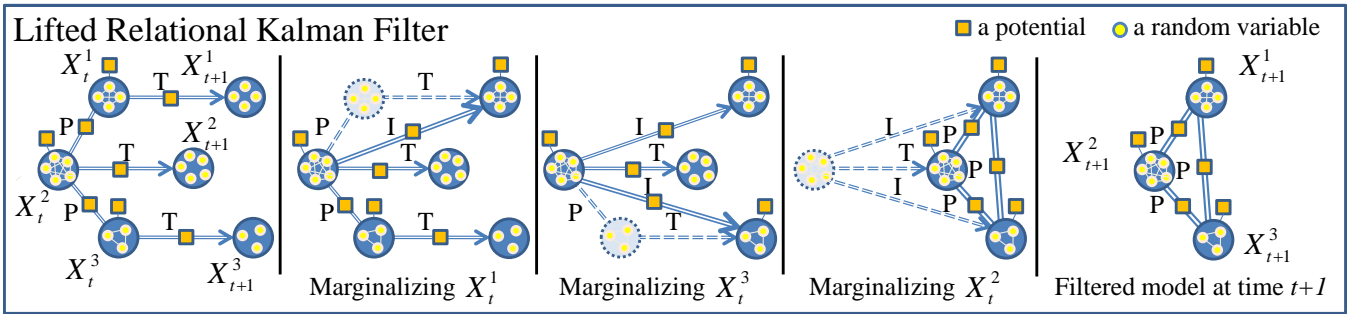


Figure 2: This model has three relational atoms, X_i , which may *represent* any number of random variables. The relational representation dramatically eliminates the need for redundant potentials. Hence, representation and filtering become much more efficient than in the propositional case. Note that the conventional KF representation is not suited for efficient (i.e. lifted) inference.

2.3 A Relational Filtering Problem

Given a prior (or current belief) over the state variables, the filtering problem is to compute the posterior after a sequence of timesteps. The input to the problem is: (1) Relational Gaussian Model (RTMs, RPMs and ROMs); (2) current belief over the relational atoms (X_t^i) represented by a product of relational Gaussian potentials; (3) sequence of control-inputs (U_t^i, \dots, U_T^i); and (4) sequence of observations (O_t^i, \dots, O_T^i). The output is the relational Gaussian posterior distribution over the relational atoms (X_T^i) at timestep T.

2.4 Input and Observation Association

At every timestep the control-inputs and observations must be associated with the random variables they affect. The ideas in this section apply to control-inputs and observations but we illustrate them for observations.

We distinguish two types of observations: direct and indirect. Direct observations are those made for a specific random variable. For instance, if we make an observation for each random variable in a subset $A_t^i \subseteq X_t^i$ of the ground substitutions of relational atom X_t^i , we are looking at the following model,

$$\prod_{a_j \in A_t^i} \phi_{ROM}(o_t^i(a_j) | X_t^i(a_j)). \quad (8)$$

In the example of Figure 1, observing the selling price of a house would dramatically reduce the variance of the hidden variable that represents the true value of that house.

Similarly, multiple direct observations, $O_t^i = \{o_t^{i,1}, o_t^{i,2}, \dots, o_t^{i,|O_t^i|}\}$, could be made for each variable in some set of random variables,

$$\prod_{a_j \in A_t^i} \prod_{o_t^{i,k} \in O_t^i} \phi_{ROM_k}(o_t^{i,k}(a_j) | X_t^i(a_j)). \quad (9)$$

Given some notion of neighborhood (e.g. a residential neighborhood or a block of houses), indirect observation allows the possibility that observations made for a random variable, $o_t^i(a')$, would influence nearby random variables, $X_t^i(a_j)$, $a' \neq a_j$,

$$\prod_{a_j \in A_t^i} \phi_{ROM}(o_t^i(a') | X_t^i(a_j)). \quad (10)$$

For example, this allows the possibility that the observation of the selling price of a house would reduce the variance of the true values of neighboring houses.

Current (exact) lifted inference algorithms (e.g. [Kersting *et al.*, 2006; Choi *et al.*, 2010]) handle observations by partitioning the relational atoms into groupings of groundings for which identical observations and observation models apply. In contrast, our approach partitions a relational atom into sets according to the number of different types of observations associated with each random variable. For instance, if an individual observation of the same ROM type is made for each random variable then no partitioning at all is necessary. The intuition for this is that the filtering process will assign the same variance to any two hidden variables for which the same number of observations is made at the current timestep.

Here, the partition will determine new RPMs, the pairwise parafactors which maintain the variances and covariances. In particular, the number of new RPMs is quadratic in the size of the partition. Since individual observations cause the means of the random variables to differ we store the mean information in the prior and posterior (P and P_{new} in Section 3). Hence, the number of priors and posteriors is linear in the number of random variables. However, this will not affect the computational complexity of inference as long as the RPMs do not degenerate. Further details are given in Sections 3.3 and 4.

Formally, given a partition $\Pi^i = (M_1^i, M_2^i, \dots, M_{|\Pi^i|}^i)$ of a relational atom, X^i , the observation model takes the form,

$$\prod_{M_l^i \in \Pi^i} \prod_{a_j \in M_l^i} \prod_{o_t^{i,k} \in O_t^i} \phi_{ROM_k}(o_t^{i,k}(a_j) | X^i(a_j)), \quad (11)$$

where we omit the time subscript and where O_t^i is the set of observations relevant to part l .

3 Lifted Relational Kalman Filter

The **Lifted Relational Kalman Filter (LRKF)**, just like the conventional Kalman Filter, carries two recursive computations: prediction step and update/correction step.

3.1 Lifted Prediction

In the prediction step, our current belief over the states of the relational atoms together with the RTMs, RPMs and control-

inputs are used to make a best estimate of state without observation information. First, the product of potentials in the RTMs and RPMs is built. Second, the variables from the previous timestep are marginalized resulting in new RPMs and estimates of the relational atoms in the current timestep. We call this estimates the *intermediate posterior*, the input to the update step.

$$\begin{aligned}
& \int_{X_t^1, \dots, X_t^n} \prod_{1 \leq i < j \leq n} \prod_{a \in A^i, a' \in A^j} \phi_{RTM}^{i,j}(X_{t+1}^i(a') | X_t^i(a), U_t^i(a)) \cdot P^i(X_t^i(a)) \cdot \phi_{RPM}^{i,j}(X_t^i(a), X_t^j(a')) \\
&= \int_{X_t^1, \dots, X_t^n} \prod_{1 \leq i < j \leq n} \prod_{a \in A^i, a' \in A^j} \exp\left(-\frac{(X_{t+1}^j(a') - B_X^{i,j} X_t^i(a) - B_U^{i,j} U_t^i(a))^2}{\sigma_{RTM}^{i,j,2}}\right) \\
&\quad \cdot \exp\left(-\frac{(X_t^i(a) - \mu_{pr}^i(a))^2}{\sigma_{pr}^{i,2}}\right) \cdot \exp\left(-\frac{(X_t^i(a) - R_X^{i,j} X_t^j(a'))^2}{\sigma_{RPM}^{i,j,2}}\right) \quad (12) \\
&= \prod_{1 \leq i < j \leq n} \prod_{a \in A^i, a' \in A^j} \exp\left(-\frac{(X_{t+1}^j(a') - R_X^{i,j} X_t^j(a'))^2}{\sigma_{RPM}^{i,j,2}}\right) \cdot \exp\left(-\frac{(X_{t+1}^i(a) - \mu_{pr}^i(a))^2}{\sigma_{pr}^{i,2}}\right) \\
&= \prod_{1 \leq i < j \leq n} \prod_{a \in A^i, a' \in A^j} \phi_{RPM}^{i,j}(X_{t+1}^i(a), X_{t+1}^j(a')) \cdot \prod_{1 \leq i \leq n} \prod_{a \in A^i} P^i(X_{t+1}^i(a)). \quad (13)
\end{aligned}$$

Here, $\phi_{RPM}^{i,j}$, P^i and $P^{i'}$ are respectively the updated RPMs, the priors and the intermediate posteriors. More details of the integration are given in Appendix A.

3.2 Lifted Update

In the update step, the intermediate posterior $P^{i'}$ and ROMs are used to correct our estimate of the relational atoms.

When a single observation, O_{t+1}^i , is associated with all variables in a relational atom, we calculate the posterior for one random variable $X_{t+1}^i(a)$ and use the result for the rest of the groundings of the same relational atom,

$$\begin{aligned}
& P^{i'}(X_{t+1}^i(a)) \cdot \phi_{ROM}(O_{t+1}^i | X_{t+1}^i(a)) \\
&= \exp\left(-\frac{(X_{t+1}^i(a) - \mu_{pr}^i(a))^2}{\sigma_{pr}^{i,2}}\right) \cdot \exp\left(-\frac{(X_{t+1}^i(a) - o_{t+1}^i)^2}{\sigma_{ROM}^{i,2}}\right) \\
&= \exp\left(\frac{-X_{t+1}^i(a)^2 + 2\mu_{pr}^i(a)X_{t+1}^i(a) - \mu_{pr}^i(a)^2}{\sigma_{pr}^{i,2}} + \frac{-X_{t+1}^i(a)^2 + 2o_{t+1}^i X_{t+1}^i(a) - o_{t+1}^i{}^2}{\sigma_{ROM}^{i,2}}\right) \\
&= c' \cdot \exp\left(-\frac{(X_{t+1}^i(a) - \mu_{pr_{new}}^i)^2}{\sigma_{pr_{new}}^{i,2}}\right) = P_{new}^i(X_{t+1}^i(a)). \quad (14)
\end{aligned}$$

In the case of multiple observations $O_{t+1}^i = o_{t+1}^{i,1}, o_{t+1}^{i,2}, \dots, o_{t+1}^{i,|O_t^i|}$ we may also do the computation of the posterior for a single random variable $X_{t+1}^i(a)$ and use the resulting posterior for all other groundings of the relational atom (to which the same set of observations applies). The calculation is similar to the above, except that multiple observations need to be considered,

$$\begin{aligned}
& P^{i'}(X_{t+1}^i(a)) \cdot \prod_{o \in O_{t+1}^i} \phi_{ROM}(o | X_{t+1}^i(a)) \\
&= \exp\left(-\frac{(X_{t+1}^i(a) - \mu_{pr}^i(a))^2}{\sigma_{pr}^{i,2}}\right) \cdot \exp\left(-\sum_{o \in O_{t+1}^i} \frac{(X_{t+1}^i(a) - o)^2}{\sigma_{ROM}^{i,2}}\right) \\
&= c'' \cdot \exp\left(-\frac{(X_{t+1}^i(a) - \mu_{pr_{new}}^i)^2}{\sigma_{pr_{new}}^{i,2}}\right) = P_{new}^i(X_{t+1}^i(a)). \quad (15)
\end{aligned}$$

3.3 Lifted Inference with Individual Observations

One of the key challenges in lifted inference is handling individual observations. Current methods ground a relational atom when different observations are made for its random variables. It is usually the case that models shatter combinatorially fast and thus forfeit the benefits of a relational representation and the applicability of lifted inference.

We solve this problem in the LRKF by noting that the variances and covariances in the model are not affected by individual observations. We are thus able to represent the variances and covariances in a relational way while allowing variables to carry individual means. Further, the lifted prediction operation applies unmodified to this representation.

Lemma 1 *The variances of two random variables $X(a)$, $X(b)$ in an RGM are equal after a filtering step (Lifted Prediction and Lifted Update) if the following conditions hold before the filtering step: (1) both random variables are in the same relational atom; (2) the variance of both variables is the same; (3) observations are made for both variables or none of them.*

Proof Given conditions (1) and (2), we first prove that the variance of both random variables is the same after the Lifted Prediction step. Note that condition (3) is not relevant to this step.

WLOG we assume $X_t(a)$ and $X_t(b)$ have different means, $\mu_t(a)$ and $\mu_t(b)$. Moreover, it is easy to see that the variance of $X_{t+1}^i(a)$ and $X_{t+1}^i(b)$ is the same after marginalizing all random variables of timestep t due to the following two reasons: (i) $X(a)$ and $X(b)$ are in the same relational atom and thus share the same relationships with other random variables; (ii) the means are not involved in the marginalizations (see Section 3.1). It follows that we can represent the potentials relevant to the marginalization of $X_t(a)$ and $X_t(b)$ as follows:

$$\begin{aligned}
& \exp\left(-\frac{(X_t(a) - \mu_t(a))^2}{\sigma_{X_t(a)}^2}\right) \cdot \phi_{RTM}(X_{t+1}(a) | X_t(a), U_t(a)) \cdot \phi_{RPM}(X_t(a), X_t(b)) \\
&\quad \cdot \exp\left(-\frac{(X_t(b) - \mu_t(b))^2}{\sigma_{X_t(b)}^2}\right) \cdot \phi_{RTM}(X_{t+1}(b) | X_t(b), U_t(b)) \cdot \phi_{RPM}(X_t(a), X_t(b)) \\
&= \exp(c_{X_t(a)} X_t(a)^2 + c_{X_t(a)} X_t(a)) \exp\left(\frac{2B_X^{i,j}}{\sigma_{RTM}^2} X_t(a) X_{t+1}(a)\right) \exp\left(\frac{X_t(a) X_t(b)}{\sigma_{RPM}^2}\right) \\
&\quad \exp(c_{X_t(b)} X_t(b)^2 + c_{X_t(b)} X_t(b)) \exp\left(\frac{2B_X^{j,i}}{\sigma_{RTM}^2} X_t(b) X_{t+1}(b)\right) \\
&\quad \cdot \phi_{other}(X_{t+1}(a), X_{t+1}(b)),
\end{aligned}$$

where c_X refers to the coefficient of the term X^4 .

After $X_t(a)$ and $X_t(b)$ are marginalized we get a potential on $X_{t+1}(a)$ and $X_{t+1}(b)$. The variances of the random variables are the inverses of the coefficients of their squares in the resulting potential. Thus, all we need to show is that the coefficients of the square of the random variables, $X_{t+1}(a)^2$ and $X_{t+1}(b)^2$, are the same after marginalization. The two coefficients can be represented as follows,

$$c_{X_{t+1}(a)^2} = \frac{-c_{X_t(b)^2} \left(\frac{B_X^{i,j}}{\sigma_{RTM}^2}\right)^2}{\left(\frac{1}{\sigma_{RTM}^2}\right)^2 - c_{X_t(a)} c_{X_t(b)}}, \quad c_{X_{t+1}(b)^2} = \frac{-c_{X_t(a)^2} \left(\frac{B_X^{j,i}}{\sigma_{RTM}^2}\right)^2}{\left(\frac{1}{\sigma_{RTM}^2}\right)^2 - c_{X_t(a)} c_{X_t(b)}}$$

⁴For the sake of exposition the RTMs here represent dependences from state variables at time t to the same state variable at time $t+1$ (e.g. from $X_t(a)$ to $X_{t+1}(a)$). However, the general RTMs (e.g. dependences from $X_t(a)$ to $X_{t+1}(b)$) produce similar forms.

where, $\mathbf{c}_{X_i(\cdot)^2} = -\left(\frac{1}{\sigma_{X_i(\cdot)}^2} + \frac{1}{\sigma_{RTM}^2} + \frac{1}{\sigma_{RPM_i}^2}\right)$.

Condition (2) ($\sigma_{X_i(a)}^2 = \sigma_{X_i(b)}^2$) implies $\mathbf{c}_{X_i(a)^2} = \mathbf{c}_{X_i(b)^2}$ which in turn implies $\mathbf{c}_{X_{t+1}(a)^2} = \mathbf{c}_{X_{t+1}(b)^2}$. This is enough to prove that the variance of two random variables $X(a)$ and $X(b)$ with different means is the same after the Lifted Prediction step.

We now prove the result for the Lifted Update step. Regarding condition (3) there are two cases: (a) observations were made for both variables; or (b) no observations were made for either variable. In the case of (b) the proof is complete. In the case of (a), the update step for $X(a)$ can be represented by,

$$\exp\left(-\frac{(X_{t+1}(a) - \mu_{X_{t+1}(a)})^2}{\sigma_{X_{t+1}(a)}^2} - \frac{(X_{t+1}(a) - o_{a_t})^2}{\sigma_{X_{ROM}}^2}\right) = \exp\left(-\frac{(X_{t+1}(a) - \mu^+_{X_{t+1}(a)})^2}{\sigma^+_{X_{t+1}(a)}^2}\right)$$

where,

$$\sigma^+_{X_{t+1}(a)} = \frac{\sigma_{X_{t+1}(a)}^2 \sigma_{X_{ROM}}^2}{\sigma_{X_{t+1}(a)}^2 + \sigma_{X_{ROM}}^2}, \quad \mu^+_{X_{t+1}(a)} = \frac{\sigma_{X_{ROM}}^2 \mu_{X_{t+1}(a)} + \sigma_{X_{t+1}(a)}^2 o_{a_t}}{\sigma_{X_{ROM}}^2 + \sigma_{X_{t+1}(a)}^2}$$

Likewise, after the update step the variance of $X(b)$ is,

$$\sigma^+_{X_{t+1}(b)} = \frac{\sigma_{X_{t+1}(b)}^2 \sigma_{X_{ROM}}^2}{\sigma_{X_{t+1}(b)}^2 + \sigma_{X_{ROM}}^2}$$

By condition (2) and the proof for the prediction step, $\sigma_{X_{t+1}(a)} = \sigma_{X_{t+1}(b)}$. Thus, $\sigma^+_{X_{t+1}(a)} = \sigma^+_{X_{t+1}(b)}$. ■

Lemma 2 *The covariances of two pairs of variables ($X(a)$, $X(b)$) and ($X(a)$, $X(c)$) in an RGM are equal after a filtering step (Lifted Prediction and Lifted Update) if the following conditions hold before the filtering step: (1) the three random variables are in the same relational atom; (2) the covariance of both pairs of variables is the same; (3) observations are made for the three variables or none of them.*

Proof The method used in the proof of Lemma 1 can be employed in this proof: The terms involving the individual observations do not affect terms which determine the covariance of two random variables. ■

4 Algorithms and Computational Complexity

Let \mathcal{X} ($|\mathcal{X}|$) be the set (number) of all random variables in the model and $X = (X^1, \dots, X^{|\mathcal{X}|})$ be the set of relational atoms (also, a partition of \mathcal{X}). In this section we speak of the relational atoms as sets of random variables.

Figure 3 presents our Lifted Kalman Filtering algorithm. The inputs to the algorithm are: relational atoms, X ; the RGM, RTMs M_X , RPMs M_P and ROMs M_O ; the prior over the relational atoms, P_0 ; and the control-inputs, $U_{[1, \dots, T]}$, and observations, $O_{[1, \dots, T]}$, for each timestep.

The algorithm computes the posterior recursively. **Split** partitions the domains of each relational atom X^i as induced by the control-inputs U_t . **Lifted Predict** calculates new RPMs, M_P^5 , and intermediate posterior, P_{int} , based on the transition models, M_X , and the control-inputs, U_t . Then,

⁵In our representation the number of relational atoms determines the number of RPMs which is equal to $E(|X|, 2)$ (the number of 2-combinations of $|X|$ with repetition).

Split_Obs partitions the domains of each relational atom X^i as induced by the observations, O_t^i . **Lifted Update** calculates the new posterior, P_{cur} , based on the intermediate posterior, P_{int} , the observation models, M_O , and the observations, O_t^i .

Given the control-inputs, **Split** partitions relational atoms as done in previous work: e.g. *Split* [Poole, 2003] and *SHATTER* [de Salvo Braz *et al.*, 2005]. If the control-inputs are allowed to differ for the variables in a relational atom, the model will be propositionalized. Hence, there is little advance in how we handle individual control-inputs with respect to previous algorithms [Choi *et al.*, 2010].⁶

Algorithm **Split_Obs** partitions a relational atom X^i based on the observations. However, **Split_Obs** will only partition a relational atom in case the conditions of Lemmas 1 and 2 do not hold, i.e., when different number of observations are made for the relational variables. If the conditions of Lemmas 1 and 2 hold, the efficiency of the relational representation will be preserved even if multiple observations are made for all variables in some or all of the relational atoms.

```

PROCEDURE LRKF( $X, M_X, M_P, M_O, P_0, U_{[1, \dots, T]}, O_{[1, \dots, T]}$ )
  Atoms,  $X = (X^1, \dots, X^{|\mathcal{X}|})$ ; RTM,  $M_X$ , RPM,  $M_P$ , and ROM  $M_O$ ;
  prior,  $P_0$ ; control-inputs,  $U_{[1, \dots, T]}$ ; observations,  $O_{[1, \dots, T]}$ .
  1.  $P_{cur} \leftarrow P_0, X_{cur} \leftarrow X$ 
  2. For  $t = 1$  to  $T$ 
    (a)  $[X_{cur}, M_X, M_P, M_O] \leftarrow \text{Split}(X_{cur}, U_t, M_X, M_P, M_O)$ 
    (b)  $[P_{int}, M_P] \leftarrow \text{Lifted Predict}(X_{cur}, P_{cur}, M_X, M_P, U_t)$ 
        (§3.1)
    (c)  $[X_{cur}, M_O] \leftarrow \text{Split Obs}(X_{cur}, O_t, M_O)$  (§3.3)
    (d)  $[P_{cur}] \leftarrow \text{Lifted Update}(X_{cur}, M_O, P_{int}, O_t)$  (§3.2)
  3. Return  $X_{cur}, P_{cur}$ 

```

Figure 3: Algorithm Lifted Relational Kalman Filter for Relational Gaussian Models.

Lemma 3 *The complexity of Lifted Predict is $O(|\mathcal{X}| \cdot |X_{t+}^*|^2)$. Where X_{t+}^* is the set of relational atoms output by Split.*

Proof This step corresponds to the marginalization (Equation (13) and Appendix A) of the variables in X . For every variable that is integrated the parameters of all, $E(|X_{t+}|, 2)$, pairwise interactions between relational atoms must be updated. ■

Lemma 4 *The complexity of Lifted Update is $O(|X_{t+}^*| \cdot |O_{max}|)$. Where X_{t+}^* is the set of relational atoms output by Split_Obs and O_{max} is the largest set of observations associated with a relational atom.*

Proof For each relational atom in X_{t+}^* the computation in Equation (15) iterates over all relevant observations. ■

Our main result follows,

Theorem 5 *The computational complexity of LRKF is $O(T \cdot (|\mathcal{X}| \cdot |X_{t+}^*|^2 + |X_{t+}^*| \cdot |O_{max}^*|))$ where T is the number of timesteps, X , X_{t+}^* , X_{t+}^* and O_{max}^* are as above with the * indicating the largest set across all timesteps.*

⁶However, we conjecture that techniques similar to the ones we used for ROMs can be applied to RTMs. Any two random variables in the same atom will have the same variance after the Lifted Predict step if they receive the same types of control inputs. That is, RTMs of the same type will increase the variances of the random variables by the same amount.

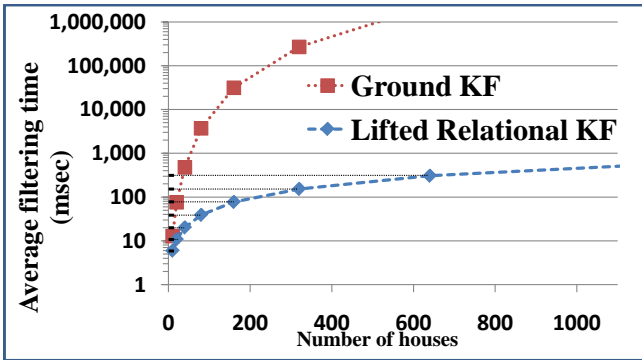


Figure 4: Average filtering time with increasing number of houses. Note the cubic increase in filtering time for the Ground Kalman Filter and the linear increase for our Lifted Relational Kalman Filter (LRKF). The y-axis is shown in logarithmic scale. To show that LRKF performs linearly, we added markers at the measurements on the LRKF curve.

5 Experimental Results

We compare the average filtering time of **LRKF** and a conventional Kalman Filter by varying the number of random variables. We implemented both the LRKF and the conventional KF (which handles random variables individually) in Perl. This makes the manipulation of the dynamically changing structure convenient.

For the housing market model in Figure 1, we randomly choose the parameters of the models (priors, RTMs, RPMs, and ROMs) and provide observations for HMO_t and $HPO_t(\cdot)$. To emphasize the difference in scalability, we assume that some set of houses has individual observations in each timestep, $HPO_t(\cdot)$, while the rest of the houses do not. We ran the two filters over 50 timesteps. The results in graph 4 confirm our theoretical results contrasting the linear time complexity of **LRKF** with the cubic time complexity of the Kalman Filter.

6 Application to Online Social Networks

For decades social scientists have studied how different types of relationships impact individuals and organizations. More recently, research on analyzing online social networks (OSNs) has demonstrated that relational patterns can be exploited to improve predictive models of link structure and behavior. Further, the accurate estimation of relationship strengths has applications to understanding human behavior, predicting human behavior (e.g. fraud), privacy control, information prioritization, recommender systems, search and visualization (see e.g. [Gilbert and Karahalios, 2009], [Xian *et al.*, 2010] and references therein).

Previous work on OSNs has been characterized by two major limitations: (1) Following social media it has focused on binary friendship relations, i.e., two peo-

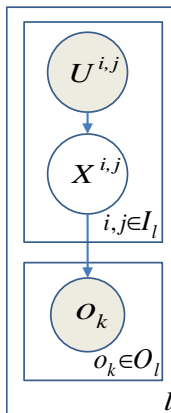


Figure 5: Model with plate notation and induced partition.

ple can be either friends or strangers. However, in reality relationships may fall anywhere along a continuous spectrum - an observation made in the social sciences since 1973 [Granovetter, 1973] with the introduction of the notion of tie strength. (2) Works on link prediction and relationship strength estimation have given little attention to the dynamic nature of social networks (SNs). However, it has been demonstrated that the level of interactions between individuals varies widely over time. For instance, experiments on the Facebook OSN showed that, on average, 55% of the links that are active during a given month are no longer active during the following month [Viswanath *et al.*, 2009].

Given the relational nature of SNs and the fact that interaction data is notably noisy a probabilistic and relational approach to prediction is best. However, predicting relationship strengths on a single network-snapshot has remained too expensive for current exact inference algorithms [Sen *et al.*, 2008]. To exacerbate the above limitations, current OSNs are potentially very large and rapidly growing.

Here we show that our LRKF is able to carry accurate relationship strength prediction on large dynamic networks while using state-of-the-art modeling features and techniques. Following the modeling decisions made by [Xian *et al.*, 2010] we propose a probabilistic model where hidden variables represent relationship strength. The model is composed of two parts (Figure 5): (1) A generative component (top) models the conditional probability of relationship strength given profile similarities and; (2) a discriminative component (bottom) models the conditional probability of the interaction activity between users given the strength of their relationship.

In [Xian *et al.*, 2010] the hidden variables are estimated iteratively given a snapshot of the OSN. In sharp contrast, we are able to filter the state of the network as observations about user interactions are made. We extend previous work by introducing a linear Gaussian model of relationship evolution (Equation (16)) that relates state variables across timesteps. For this purpose, the control-inputs are derived from a linear-combination of profile similarities. The model has the following components,

$$\begin{aligned}
 P(X_t(i, j)) &= \exp\left(-\frac{(X_t(i, j) - c_t(i, j))^2}{2\sigma_t^2}\right) \\
 \prod_{i \in \mathcal{U}} \phi_{U_i}(X_t(i, j), U_t(i, j)) &= \prod_{i \in \mathcal{U}} \exp\left(-\frac{(X_t(i, j) - \alpha_{U_i} - \beta_{U_i} U_t(i, j))^2}{2\sigma_{U_i}^2}\right) \\
 \prod_{o_k^k \in \mathcal{O}_{t+1}} \phi_{ROM_k}(o_{t+1}^k, X_{t+1}(i, j)) &= \prod_{o_k^k \in \mathcal{O}_{t+1}} \exp\left(-\frac{(o_{t+1}^k - \alpha_{O_k} - \beta_{O_k} X_{t+1}(i, j))^2}{2\sigma_{O_k}^2}\right) \\
 \phi_{RTM}(X_{t+1}(i, j), X_t(i, j), U_t(i, j)) &= \exp\left(-\frac{(X_{t+1}(i, j) - \mathbf{w} \cdot \mathbf{U}_t(i, j) - \beta_T X_t(i, j))^2}{2\sigma_T^2}\right)
 \end{aligned} \tag{16}$$

where (i, j) ranges over all pairs of individuals, X are the relationship strength variables, \mathbf{U} are the profile similarity feature vectors and the o^k are interaction observations.

One limitation of this model, which it inherits from its static predecessor, is that relationship strengths are considered independent of each other. A better approach is to consider the dependences between edges by introducing RPMs of the form,

$$\phi_{RPM}(X_t(i, j), X_t(i, k)). \tag{17}$$

We envision a multivariate Gaussian version of probabilistic transitivity being represented in this fashion.

7 Related Work

The KF [Kalman, 1960; Roweis and Ghahramani, 1999] is a method for estimating the state of a dynamic process given a sequence of noisy observations. It is restricted to linear dynamic and linear measurement models both with additive Gaussian noise. The Extended Kalman Filter (EKF) [Sorenson and Stubberud, 1968] extends the KF to non-linear systems. For high dimensional data, a sampling method has been devised, the Ensemble Kalman Filter [Evensen, 1994]. Exact Kalman Filtering for high dimensional data is not feasible because exact filtering requires matrix inversions which take time cubic in the number of random variables.

Our RGMs represent the probability density as a product of node and edge factors. Any multivariate Gaussian is a quadratic exponential and can thus be written in this form. This is related to the information form of the Gaussian density and is the basis of other models such as Directed Gaussian Models (DGMs) [Cowell, 1998] and Gaussian Markov Random Fields (GMRFs) [Rue and Held, 2005]. However, RGMs are relational while DGMs and GMRFs are not. Thus, the previous models do not have a compact (relational) representations and, more importantly, an efficient (lifted) exact inference algorithm.

Relational probabilistic models allow the specification of models with size independent of the sizes of the populations in the model [Friedman *et al.*, 1999; Poole, 2003; Richardson and Domingos, 2006]. Lifted inference algorithms [de Salvo Braz *et al.*, 2005; Milch and Russell, 2006] attempt to carry as much of the computations without propositionalizing the model. [Poole, 2003], solves inference problems by dynamically splitting and unifying sets of ground atoms. [de Salvo Braz *et al.*, 2005] (FOVE) introduced *counting elimination* to efficiently eliminate atoms with different parameterizations. [Milch *et al.*, 2008] (C-FOVE) take a slightly different approach with the introduction of counting formulas. However, all of the above lifted inference algorithms are not applicable to models with continuous variables.

[Kersting *et al.*, 2006] introduced Logical HMMs that combine ideas from *Statistical Relational Learning* and dynamic models. Indeed their work, as ours, pursues the benefits that the relational approach brings to inference and learning. However, their work is inherently discrete and further, they assume specific transition and observation models.

For relational models with continuous variables, recent advances have made inference possible. [Wang and Domingos, 2008] is an approximate algorithm based on sampling, search and local optimization. [Choi *et al.*, 2010] is an exact variable elimination algorithm for continuous domains. The latter algorithm is similar to the marginalization problem that is part of the prediction step in filtering. However, none of these algorithms have been devised with dynamic models in mind nor do they address the problem of individual observations.

8 Conclusion and Future Work

We propose Relational Gaussian Models to represent and model dynamic systems in a relational (first-order) way. Fur-

ther, we present the first algorithm for filtering or tracking at the first-order level. Our theoretical analysis and empirical tests show that our approach leads to significant gains in efficiency and enables filtering for systems with very large numbers of random variables. We also make the case for the applicability of lifted inference to address real-world problems by taking a recently proposed model of social relationship strength and extending it to large dynamic networks.

A limitation of our exact filtering is that we shatter the model when the random variables in a relational atom receive different numbers of observations because their variances and covariances become different. Our current understanding is that approximate re-grouping of random variables is the only general recourse in this case.

Acknowledgements

We wish to thank Jihye Seong, Jeongkeun Lee and the anonymous reviewers for their valuable comments. This work was supported in part by NSF award IIS-09-17123 – *RI: Scaling Up Inference in Dynamic Systems with Logical Structure* and NSF award ECS-09-43627 – *Improving Prediction of Subsurface Flow and Transport through Exploratory Data Analysis and Complementary Modeling*. Abner Guzman-Rivera was supported by the C2S2 Focus Center, one of six research centers funded under FCRP, a Semiconductor Research Corp. entity.

A Details of Lifted Prediction

The integration is done using the following rule,

$$\int_{X_t^i(a^*)} \exp(-\mathbb{A}X_t^i(a^*)^2 + 2\mathbb{B}X_t^i(a^*) - \mathbb{C}) = \frac{\sqrt{\pi}}{\sqrt{\mathbb{A}}} \exp\left(\frac{\mathbb{B}^2}{\mathbb{A}} - \mathbb{C}\right). \quad (18)$$

where \mathbb{A} is a constant, \mathbb{B} a linear form of random variables except $X_t^i(a^*)$, and \mathbb{C} is a quadratic form of random variables except $X_t^i(a^*)$.

The integration of one random variable in Equation (12) can be represented as follows,

$$\int_{X_t^i(a^*)} \prod_{1 \leq j \leq n} \prod_{a' \in A^j} \exp\left(-\frac{(X_{t+1}^j(a') - B_X^{ij} X_t^i(a^*) - B_U^{ij} U_t^i(a^*))^2}{\sigma_{RTM}^i}\right) \exp\left(-\frac{(X_t^i(a^*) - \mu_p^i(a^*))^2}{\sigma_p^i}\right) \cdot \exp\left(-\frac{(X_t^i(a^*) - R_t^{ij} X_t^j(a'))^2}{\sigma_{RPM}^{ij}}\right) \quad (19)$$

$$\begin{aligned} &= \int_{X_t^i(a^*)} \exp\left(-\mathbb{A}X_t^i(a^*)^2 + \left(\mathbf{c} + \sum_{1 \leq j \leq n} \sum_{a' \in A^j} \mathbf{c}_t^j X_t^j(a') + \mathbf{c}_{t+1}^j X_{t+1}^j(a')\right) X_t^i(a^*) - \mathbb{C}\right) \\ &= \int_{X_t^i(a^*)} \exp\left(-\mathbb{A}X_t^i(a^*)^2 + \left(\mathbf{c} + \sum_{1 \leq j \leq n} \mathbf{c}_t^j \sum_{a' \in A^j} X_t^j(a') + \mathbf{c}_{t+1}^j \sum_{a' \in A^j} X_{t+1}^j(a')\right) X_t^i(a^*) - \mathbb{C}\right) \\ &= \int_{X_t^i(a^*)} \exp\left(-\mathbb{A}X_t^i(a^*)^2 + \left(\mathbf{c} + \sum_{1 \leq j \leq n} \mathbf{c}_t^j \mathbb{X}_t^j + \mathbf{c}_{t+1}^j \mathbb{X}_{t+1}^j\right) X_t^i(a^*) - \mathbb{C}\right), \quad (20) \end{aligned}$$

when \mathbf{c} , \mathbf{c}_t^j and \mathbf{c}_{t+1}^j represent constants calculated from Equation (19), and \mathbb{X}_t^j represents $\sum_{a' \in A^j} X_t^j(a')$.

Note the quadratic form in Equation (18) includes the following types of expression,

$$(\mathbb{X} + \mathbb{X}')^2 = [\mathbb{X}^2] + 2[\mathbb{X}\mathbb{X}'] + 2\mathbb{X}\mathbb{X}' + [\mathbb{X}'^2] + 2[\mathbb{X}'\mathbb{X}], \quad (21)$$

where $[X^2]$ is $\sum_{a \in A} X(a)^2$, and $[XX]$ is $\sum_{a, a' \in A, a \neq a'} X(a)X(a')$.
Now, Equation (20) is integrated as follows,

$$\begin{aligned}
& \frac{\sqrt{\pi}}{\sqrt{A}} \exp\left(\frac{1}{A} \left(c + \sum_{1 \leq j \leq n} c_j^i X_j^i + c_{t+1}^j X_{t+1}^j \right)^2 - C\right) \\
&= \frac{\sqrt{\pi}}{\sqrt{A}} \exp\left(\frac{1}{A} \sum_{1 \leq j \leq n} \left(c_j^i{}^2 [X_j^i{}^2] + 2c_j^i{}^2 [X_j^i X_j^i] + 2cc_j^i X_j^i \right)\right) \cdot \exp(-C) \\
&\cdot \exp\left(\frac{1}{A} \sum_{1 \leq j \leq n} \left(c_{t+1}^j{}^2 [X_{t+1}^j{}^2] + 2c_{t+1}^j{}^2 [X_{t+1}^j X_{t+1}^j] + 2cc_{t+1}^j X_{t+1}^j \right)\right) \\
&\cdot \exp\left(\frac{1}{A} \sum_{1 \leq j < j' \leq n} \left(2c_j^i c_{t+1}^j X_j^i X_{t+1}^j + 2c_{t+1}^j c_{t+1}^{j'} X_{t+1}^j X_{t+1}^{j'} + 2c_{t+1}^j c_{t+1}^{j'} X_{t+1}^j X_{t+1}^{j'} \right)\right) \quad (22) \\
&= \prod_{1 \leq j < j' \leq n} \prod_{\substack{a \in A^j, a' \in A^{j'} \\ a \neq a' \text{ if } i=j \\ a' \neq a' \text{ if } i=j'}} \exp\left(-\frac{(X_t^j(a) - R_{t+1}^{jj'} X_{t+1}^{j'}(a'))^2}{\sigma_{t+1RPM}^{jj'}{}^2}\right) \exp\left(-\frac{(X_{t+1}^j(a) - \mu_{t+1}^{j'}(a))^2}{\sigma_{t+1pr}^{j'}{}^2}\right) \\
&\cdot \prod_{1 \leq j < j' \leq n} \prod_{a \in A^j, a' \in A^{j'}} \exp\left(-\frac{(X_{t+1}^j(a) - R_{t+1}^{jj'} X_{t+1}^{j'}(a'))^2}{\sigma_{t+1RPM}^{jj'}{}^2}\right) \exp\left(-\frac{(X_{t+1}^j(a) - \mu_{t+1}^{j'}(a))^2}{\sigma_{t+1pr}^{j'}{}^2}\right) \\
&\cdot \prod_{1 \leq j < j' \leq n} \prod_{\substack{a \in A^j, a' \in A^{j'} \\ a' \neq a' \text{ if } i=j'}} \exp\left(-\frac{(X_{t+1}^j(a) - R_{t+1}^{jj'} X_{t+1}^{j'}(a'))^2}{\sigma_{t+1RPM}^{jj'}{}^2}\right).
\end{aligned}$$

Here, R'_t , R'_{t+1} , R'_{t+1} , μ_t , μ_{t+1} , σ'_{t+1RPM} and σ'_{t+1pr} are new constants derived from Equation (22).

References

- [Bahmani-Oskooee and Brown, 2004] Mohsen Bahmani-Oskooee and Ford Brown. Kalman filter approach to estimate the demand for international reserves. *Applied Economics*, 36(15):1655–1668, 2004.
- [Burgers *et al.*, 1998] Gerrit Burgers, Peter Jan van Leeuwen, Geir Evensen, Gerrit Burgers, and Gerrit Burgers. On the analysis scheme in the ensemble kalman filter. *Monthly Weather Review*, 126:1719–1724, 1998.
- [Choi *et al.*, 2010] Jaesik Choi, David J. Hill, and Eyal Amir. Lifted inference for relational continuous models. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 126–134, 2010.
- [Cowell, 1998] Robert Cowell. Advanced inference in bayesian networks. In Michel I. Jordan, editor, *Learning in graphical models*, pages 27–49. MIT Press, Cambridge, MA, 1998.
- [de Salvo Braz *et al.*, 2005] Rodrigo de Salvo Braz, Eyal Amir, and Dan Roth. Lifted first-order probabilistic inference. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1319–1325, 2005.
- [Evensen, 1994] Geir Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. *Journal of Geophysical Research*, 99:10143–10162, 1994.
- [Friedman *et al.*, 1999] Nir Friedman, Lise Getoor, Daphne Koller, and Avi Pfeffer. Learning probabilistic relational models. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1300–1309, 1999.
- [Gilbert and Karahalios, 2009] Eric Gilbert and Karrie Karahalios. Predicting tie strength with social media. In *Proceedings of the Conference on Human Factors in Computing Systems*, pages 211–220, 2009.
- [Granovetter, 1973] Mark S. Granovetter. The strength of weak ties. *The American Journal of Sociology*, 78(6):1360–1380, 1973.
- [Kalman, 1960] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D):35–45, 1960.
- [Kersting *et al.*, 2006] K. Kersting, L. De Raedt, and T. Raiko. Logical hidden markov models. *Journal of Artificial Intelligence Research*, 25:425–456, 2006.
- [Limketkai *et al.*, 2005] Benson Limketkai, Lin Liao, and Dieter Fox. Relational object maps for mobile robots. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1471–1476, 2005.
- [Milch and Russell, 2006] Brian Milch and Stuart J. Russell. First-order probabilistic languages: Into the unknown. In *Proceedings of the International Conference on Inductive Logic Programming*, pages 10–24, 2006.
- [Milch *et al.*, 2008] Brian Milch, Luke S. Zettlemoyer, Kristian Kersting, Michael Haimes, and Leslie Pack Kaelbling. Lifted probabilistic inference with counting formulas. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1062–1068, 2008.
- [Poole, 2003] David Poole. First-order probabilistic inference. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 985–991, 2003.
- [Richardson and Domingos, 2006] Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine Learning*, 62(1-2), 2006.
- [Roweis and Ghahramani, 1999] Sam Roweis and Zoubin Ghahramani. A unifying review of linear gaussian models. *Neural Computation*, 11(2), 1999.
- [Rue and Held, 2005] H. Rue and L. Held. *Gaussian Markov random fields: Theory and applications*. Springer, New York, 2005.
- [Sen *et al.*, 2008] Prithviraj Sen, Galileo Mark Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.
- [Sorenson and Stubberud, 1968] H. W. Sorenson and A. R. Stubberud. Non-linear filtering by approximation of the a posteriori density. *International Journal of Control*, 8:33–51, 1968.
- [Viswanath *et al.*, 2009] Bimal Viswanath, Alan Mislove, Meeyoung Cha, and Krishna P. Gummadi. On the evolution of user interactions in facebook. In *Proceedings of the ACM Workshop on Online Social Networks*, pages 37–42, 2009.
- [Wang and Domingos, 2008] Jue Wang and Pedro Domingos. Hybrid markov logic networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1106–1111, 2008.
- [Xian *et al.*, 2010] Rogjing Xian, Jennifer Neville, and Monica Rogati. Modeling relationship strength in online social networks. In *Proceedings of the International Conference on World Wide Web*, pages 981–990, 2010.