Parameter Estimation for Relational Kalman Filtering

Jaesik Choi
School of Electrical and Computer Engineering
Ulsan National Institute of Science and Technology
Ulsan, Korea

Eyal Amir
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL, USA

Tianfang Xu and **Albert J. Valocchi**
Department of Civil and Environmental Engineering
University of Illinois at Urbana-Champaign
Urbana, IL, USA

Abstract

The Kalman Filter (KF) is pervasively used to control a vast array of consumer, health and defense products. By grouping sets of symmetric state variables, the Relational Kalman Filter (RKF) enables to scale the exact KF for large-scale dynamic systems. In this paper, we provide a parameter learning algorithm for RKF, and a regrouping algorithm that prevents the degeneration of the relational structure for efficient filtering. The proposed algorithms significantly expand the applicability of the RKFs by solving the following questions: (1) how to learn parameters for RKF in partial observations; and (2) how to regroup the degenerated state variables by noisy real-world observations. We show that our new algorithms improve the efficiency of filtering the large-scale dynamic system.

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 variables in a linear dynamic system with Gaussian noise given a sequence of control-inputs and observations. The KF has been applied in a broad range of domains such as robotics, finance (Bahmani-Oskooee and Brown 2004) and environmental science (F.P. and Bierkens 2001; Clark et al. 2008). Given a sequence of observations and linear dependences with Gaussian noise between variables, the KF calculates the conditional probability density of the state variables at each time step.

Unfortunately, the KF computations are cubic in the number of state variables, which limits the use of existing exact methods to domains with a large number of state variables. This has led to the combination of approximation and sampling in the Ensemble Kalman Filter (Evensen 1994), and recently to the Relational Kalman Filters (RKFs) over grouped state variables (Choi, Guzman-Rivera, and Amir 2011; Ahmadi, Kersting, and Sanner 2011). The RKFs leverage the ability of relational languages to specify models with size of representation independent of the size of populations involved (Friedman et al. 1999; Poole 2003; Richardson and Domingos 2006).

Lifted inference algorithms for relational continuous models (Wang and Domingos 2008; Choi, Hill, and Amir 2010; Ahmadi, Kersting, and Sanner 2011) degenerate (or split) relational structures upon individual observations. Lifted RKF (Choi, Guzman-Rivera, and Amir 2011) maintains relational structure when the same number of observations are made. Otherwise, it also degenerates (possibly rapidly) the relational structure, thus lifted RKF may not be useful with sparse observations.

The main contributions of this paper are (1) to regroup the degenerated state variables from noisy real-world observations with a tight error bounds; and (2) to learn parameters for RKFs.

We propose a new learning algorithm for RKFs. We show that the relational learning expedites filtering, and achieves accurate filtering in theory and practice. The key intuition is that the Maximum Likelihood Estimate (MLE) of RKF parameters is the empirical mean and variance over state variables included in each group. For partial observations, the parameters can be calculated similarly. We show that under reasonable conditions variances of degenerated state variables on partial observations converge exponentially. Thus, our approximate regrouping algorithm has bounded errors compared to the exact KF. We could show that the RKF with regrouping is more robust against degeneracy than the Lifted RKF in practice with partial observations.

2 Relational Linear Dynamic Systems

In this section, we define relational linear dynamic systems. Dependencies among variables are represented using **relational atoms**, or just **atoms**.[1] The relational atoms are useful when the joint probability of variables involves common types of functions. When representing the joint probability

[1] For comprehensive definitions, see (Poole 2003; de Salvo Braz, Amir, and Roth 2005)

distribution, there are products of the parameterized functions (or potentials).

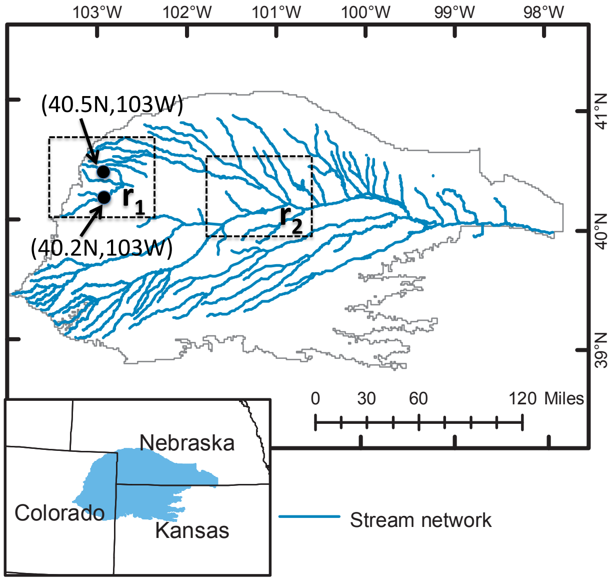

Figure 1: Republican River Basin covering portions of eastern Colorado, northwest Kansas and southwest Nebraska. This figure shows two clustered water wells; *region*$_1(r_1)$ and *region*$_2(r_2)$. Water wells in each region have the same (linear Gaussian) relationships with wells in other regions.

Relational atoms represent the set of state variables corresponding to all ground substitutions of its parameter variables. For example, let $X^{r_1}(\textit{Latitude}, \textit{Longitude})$ be an atom for the (water level of) wells in *region*$_1$, θ=$(40.2N, 103W)$. When we substitute *Latitude* and *Longitude* with θ, the atom becomes a state variable $X^{r_1}(40.2N, 103W)$ which represents the level (or prediction) of well head at $(\textit{Latitude}{=}40.2N, \textit{Longitude}{=}103W)$. Formally, applying a substitution θ to an atom $\mathbf{X}(L)$ yields a new atom $\mathbf{X}(L\theta)$ where $L\theta$ is obtained by renaming the parameter variables in L according to θ. If θ is a ground substitution, $\mathbf{X}(L\theta)$ is a ground state variable like $X^{r_1}(40.2N, 103W)$.[2] $|X(L)|$ or just $|X|$ denotes the the number of distinct state variables generated from X by all substitutions.

A pairwise Gaussian parfactor $((\mathbf{X}, \mathbf{X}'), \phi)$ is composed of a pair of two atoms ($\mathbf{X}$, $\mathbf{X}'$) and a linear Gaussian potential ϕ between two atoms in the following form.

$$\phi(\mathbf{X}, \mathbf{X}') \propto \exp\left[\frac{(\mathbf{X} - \mathbf{X}' - \mu)^2}{\sigma^2}\right]$$

For example, a pairwise Gaussian parfactor $\phi_{r1,r2}(X_{r1}, X_{r2})$ represents the linear Gaussian relationship between two ground variables chosen from *region*$_1$ and *region*$_2$ respectively.

A pairwise Gaussian factor, or just a **factor**, $f = ((\mathbf{x}, \mathbf{x}'), \phi)$ is a pair where ϕ is a potential function on $(\mathbf{x}, \mathbf{x}')$ from $\mathbb{R}^2$ to $\mathbb{R}^+$ where $(\mathbf{x}, \mathbf{x}')$ is a pair of ground random variables derived by ground substitutions from $(\mathbf{X}(L\theta), \mathbf{X}'(L'\theta'))$. A factor f defines a weighting function on a **valuation** $(x, x') = (v, v')$: $w_f(x, x') = \phi(v, v')$. The weighting function for a *parfactor* g is the product of the weighting functions over all of its ground substitutions (factors), $w_g(v) = \prod_{f \in g} w_f(v)$. Hence, a set of parfactors G defines a probability density,

$$w_G(v) = \frac{1}{Z} \prod_{g \in G} \prod_{f \in g} w_f(v),$$

where Z is the normalizing constant.[3] In this way, we can represent the joint probability of all random variables (e.g. all wells in *region*$_1$ and *region*$_2$).

Relational Transition Models (RTMs) characterize the dependence of relational atoms between time steps. $X^i_t(a)$ and $X^j_{t+1}(a')$ are relational atoms at time step t and $t+1$ respectively when a and a' are ground substitutions, e.g., θ'=(40.2N,98W). $U^i_t(a)$ is the control-input information. A RTM takes the following form,

$$X^j_{t+1}(a') = B^{ij}_X X^i_t(a) + B^{ij}_U U^i_t(a) + G^{ij}_{RTM}, \qquad (1)$$

where $G^{ij}_{RTM} \sim \mathcal{N}(0, {\sigma^{ij}_{RTM}}^2)$ and $\mathcal{N}(m, \sigma^2)$ is the normal distribution with mean m and variance σ^2. B^{ij}_X and B^{ij}_U are the linear transition coefficients.

In the linear Gaussian representation, the transition models take the following form,

$$\phi_{RTM}(X^j_{t+1}(a') | X^i_t(a), U^i_t(a)) \propto \exp\left[-\frac{(X^j_{t+1}(a') - B^{ij}_X X^i_t(a) - B^{ij}_U U^i_t(a))^2}{2{\sigma^{ij}_{RTM}}^2}\right]. (2)$$

The most common transition is the one from the state $X^i_t(a)$ to the state itself $X^i_{t+1}(a)$ at the next time step,

$$X^i_{t+1}(a) = B^i_X X^i_t(a) + B^i_U U^i_t(a) + G^i_{RTM}. \qquad (3)$$

Relational Observation Models (ROMs) represent the relationships between the hidden (state) variables, $X^i_t(a)$, and the observations made directly on the related variable, $O^i_t(a)$ (*direct observations*),

$$O^i_t(a) = H^i_t X^i_t(a) + G^i_{ROM}, G^i_{ROM} \sim \mathcal{N}(0, {\sigma^i_{ROM}}^2) \quad (4)$$

H^i_t is the linear coefficient from $X^i_t(a)$ to $O^i_t(a)$.

ROMs also represent the relationships between the hidden variables $X^i_t(a)$ and the observations made indirectly on a variable in the atom $O^i_t(a')$ where $a \neq a'$ (*relational observations*),

$$O^i_t(a') = H'^i_t X^i_t(a) + G'^i_{ROM}, G'^i_{ROM} \sim \mathcal{N}(0, {\sigma'^i_{ROM}}^2) \quad (5)$$

[2] Here, we assume that the ground state variables are univariate, e.g., domain of x is $\mathbb{R}$. However, it is straightforward to extend this model to the one with multivariate ground variables.

[3] The codition of being a probability density is that at least a random variable has a prior distribution, see (Choi, Hill, and Amir 2010).

in most cases, it is reasonable to set the variance of direct observation $var(G^{i}_{ROM})$ is smaller than the variance of relational one $var(G'^{i}_{ROM})$ s.t. $\sigma^{i}_{ROM} \ll \sigma'^{i}_{ROM}$. For the well example, $X^{r_1}_t(\mathbf{40.2N}, 103W)$, an observation made at the exact location $O^{r_1}_t(\mathbf{40.2N}, 103W)$ will have the smaller variance (more certain) than an observation made at a nearby location $O^{r_1}_t(\mathbf{40.5N}, 103W)$.

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

$$\phi_{ROM}(O^i_t(a)|X^i_t(a)) \propto \exp\left[-\frac{(O^i_t(a) - H^i_t X^i_t(a))^2}{2{\sigma^i_{ROM}}^2}\right].$$

Relational Pairwise Models (RPMs) represent linear dependences between pairs of relational atoms,

$$X^i_t(a) = R^{ij}_t X^j_t(a') + G^{ij}_{RPM}, G^{ij}_{RPM} \sim \mathcal{N}(0, \sigma^{ij}_{RPM}) \tag{6}$$

R^{ij}_t is the coefficient.

Note that RTMs and ROMs represent the nature of dynamic systems (e.g. the state at the next time step depends on the current time step). The product of RPMs is an efficient way to represent the relational structure over atoms, groups of state variables.

Relational Kalman Filter (RKF) is a filtering procedure with a relational linear dynamic system which is composed of RTMs, ROMs and RPMs. That is, the joint probability of state variables are represented by the product of pairwise Gaussian parfactors. **Lifted RKF** computes the posterior of the state variables given a prior (current) belief and full or partial observations. The input to the problem is: (1) relational parfactors (RTMs, RPMs and ROMs); (2) current belief over atoms (X^i_0); (3) a sequence of control-inputs ($U^i_1, \ldots, U^i_T$); and (4) a sequence of observations ($O^i_1, \ldots, O^i_T$). The output is the multivariate Gaussian distribution over the atoms (X^i_T) at each time step T. The filtering problem is solved by algorithms represented (Choi, Guzman-Rivera, and Amir 2011; Ahmadi, Kersting, and Sanner 2011). Here, we turn our focus to the parameter learning problem.

3 Learning Relational Kalman Filter

The two important parameters of the RKF are the transition models and observation models. In this section, we present a learning algorithm that derives the maximum likelihood estimates of RTMs and ROMs. For simplicity, we will present a solution with fully observed model, first. A solution for partial observations can be derived with a slight modification.

3.1 Algorithm *LearningRKF*

Algorithm *LearningRKF* estimates the parameter of RKF given a sequence of observations such as measurements of water wells for several years. The overall procedure is similar to the one with the parameter learning for the ground KF. Here, the main difference is that the coefficients and covariances of RTMs and ROMs are the block matrices. A subroutine, *BlockAverage*, averages out the diagonal and non-diagonal entries of an input matrix, and then ouputs a block matrix where each block includes the empirical means, variances and covariances in each block. In the following sections, we will show that the block matrix computed by *BlockAverage* is the MLE estimate.

Algorithm 1 LearningRKF

```
input: a sequence of obs $(O_1, \cdots, O_T)$
$(\mathbf{B}, \Sigma_T, \mathbf{C}, \Sigma_O) \leftarrow (I, I, I, I)$
currentLL $\leftarrow \infty$
repeat
    prevLL $\leftarrow$ currentLL
    $(\mathbf{B}', \Sigma'_T, \mathbf{C}', \Sigma'_O) \leftarrow$LearnGroundTM($O_t$,$\mathbf{B}$,$\Sigma_T$,$\mathbf{C}$,$\Sigma_O$)
    $(\mathbf{B}, \Sigma_T, \mathbf{C}, \Sigma_O) \leftarrow$ BlockAverage($\mathbf{B}', \Sigma'_T, \mathbf{C}', \Sigma'_O$)
    currentLL $\leftarrow \sum_t \log P(O_t|X_t, \mathbf{B}, \Sigma_T, \mathbf{C}, \Sigma_O)$
until | prevLL - currentLL | $< \epsilon$
output: estimated parameters $(\mathbf{B}, \Sigma_T, \mathbf{C}, \Sigma_O)$
```

3.2 Learning Transition Models

Here, we derive the parameter of the RTMs: linear coefficient B and Gaussian noise G_{RTM}.

Learning Transition Noise is to calculate the mean and the covariance matrix in the following block forms,

$$\mu_T = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix}, \Sigma_T = \begin{bmatrix} \Sigma^{1,1} & \Sigma^{1,2} & \cdots & \Sigma^{1,n} \\ \Sigma^{2,1} & \Sigma^{2,2} & \cdots & \Sigma^{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma^{n,1} & \Sigma^{n,2} & \cdots & \Sigma^{n,n} \end{bmatrix} \tag{7}$$

Where μ_i is a vector of size $|X^i|$ (=n_i); $\Sigma^{i,j}$ is a matrix of size n_i by n_j.

Given a prior, a linear coefficient $\mathbf{B}$ and a sequence of full observations, we derive the estimate X_t at time step t assuming the Gaussian noise in the transition model. The MLE estimation of μ and Σ for the RTM can be derived:

$$(\mu_{Tmax}, \Sigma_{Tmax}) = \operatorname*{arg\,max}_{\mu_T, \Sigma_T} \sum_{t=2,\cdots,T} \log f_{\mathcal{N}}(\vec{X}_t; \mu_T, \Sigma_T)$$

where $\vec{X}_t = X_t - \mathbf{B}X_{t-1}$.

Proposition 1. *Given a RKF with a single atom, the maximum likelihood estimates of the Gaussian transition noise are the empirical mean, variance and covariance as follows,*

$$\mu_{MLE} = \begin{bmatrix} m \\ m \\ \vdots \\ m \end{bmatrix}, \Sigma_{MLE} = \begin{bmatrix} \sigma^2 & \sigma' & \cdots & \sigma' \\ \sigma' & \sigma^2 & \cdots & \sigma' \\ \vdots & \vdots & \ddots & \vdots \\ \sigma' & \sigma' & \cdots & \sigma^2 \end{bmatrix}$$

such that,

$$m = \frac{1}{n\bar{T}} \sum_{t=2}^{T} \sum_{a \in A} \vec{X}_t(a), \sigma^2 = \frac{1}{n\bar{T}} \sum_{t=2}^{T} \sum_{a \in A} \left(\vec{X}_t(a) - m\right)^2$$

$$\sigma' = \frac{1}{n(n-1)\bar{T}} \sum_{t=2}^{T} \sum_{\substack{a,a' \in A \\ a \neq a'}} \left(\vec{X}_t(a) - m\right)\left(\vec{X}_t(a') - m\right).$$

where $n = |\vec{X}_t(A)|$ *and* $\bar{T} = T - 1$.

Proof. The MLEs of the parameters (μ_T, Σ_T) are derived by the partial derivatives of the log likelihood,

$$\frac{\partial}{\partial \mu_T}\sum_{t=2}^{T}\log f_{\mathcal{N}}(\vec{X}_t;\mu_T,\Sigma_T)=0,$$

$$\frac{\partial}{\partial \Sigma_T}\sum_{t=2}^{T}\log f_{\mathcal{N}}(\vec{X}_t;\mu_T,\Sigma_T)=0.$$

All ground variables generated from the atom $\vec{X}_t$ have the same mean, variance and covariances as shown in Equation (8). Now, we can specify the following linear constraints;

$$m=\frac{1}{\bar{T}}\sum_t \vec{X}_t(a_1)=\cdots=\frac{1}{\bar{T}}\sum_t \vec{X}_t(a_n).$$

That is, $m=\frac{1}{n\bar{T}}\sum_t\sum_a \vec{X}_t(a)$.

The covariance matrix of the RTM is also calculated from the empirical covariance matrix. The diagonal entries σ^2 are derived for the variances;

$$\sigma^2=\frac{1}{\bar{T}}\sum_t(\vec{X}_t(a_1)-m)^2=\cdots=\frac{1}{\bar{T}}\sum_t(\vec{X}_t(a_{n_i})-m)^2$$

Thus, $\sigma^2=\frac{1}{n\bar{T}}\sum_t\sum_a\left(\vec{X}_t(a)-m\right)^2$. Non-diagonal entries (covariances) are derived similarly with $n(n-1)$ empirical covariances.[4] □

This result is consistent with the result in non-relational KF because the MLE estimates of the ground KF (μ_T and Σ_T) are known to be the empirical mean and the empirical covariance matrix (Roweis and Ghahramani 1999).

In the general case, when we have multiple atoms, the mean vector and the covariance matrix are block forms as shown in Equation 7. That is, the mean and covariance values are same in each subblock. In case of, two atoms X^i and X^j, the means and covariances are as follows:

$$\mu_T=\begin{bmatrix}\mu_1\\ \mu_2\end{bmatrix},\Sigma_T=\begin{bmatrix}\Sigma_{1,1} & \Sigma_{1,2}\\ \Sigma_{2,1} & \Sigma_{2,2}\end{bmatrix}$$

The MLE parameters of the RTM are derived similarly with empirical means and covariances of subblocks.

Proposition 2. *Given a RKF with multiple atoms, the maximum likelihood estimates of the Gaussian transition noise are the empirical means, variances and covariances,*

$$\mu_i=[m_i,\cdots,m_i]^T\ \textit{s.t.}\ m_i=\frac{1}{n_i\bar{T}}\sum_{t=2}^{T}\sum_a \vec{X}_t^{\,i}(a),$$

$$\Sigma^{i,i}=\begin{bmatrix}\sigma^2 & \sigma' & \cdots & \sigma'\\ \sigma' & \sigma^2 & \cdots & \sigma'\\ \vdots & \vdots & \ddots & \vdots\\ \sigma' & \sigma' & \cdots & \sigma^2\end{bmatrix}\Sigma^{i,j}=\begin{bmatrix}\sigma'' & \cdots & \sigma''\\ \vdots & \ddots & \vdots\\ \sigma'' & \cdots & \sigma''\end{bmatrix}$$

$$\sigma^2\ =\ \frac{1}{n_i\bar{T}}\sum_{t=2}^{T}\sum_a\left(\vec{X}_t^i(a)-m_i\right)\left(\vec{X}_t^i(a)-m_i\right)$$

$$\sigma'\ =\ \frac{1}{n_i(n_i-1)\bar{T}}\sum_{t=2}^{T}\sum_{\substack{a,a'\in A\\(a\neq a')}}\left(\vec{X}_t^i(a)-m_i\right)\left(\vec{X}_t^i(a')-m_i\right),$$

$$\sigma''\ =\ \frac{1}{n_in_j\bar{T}}\sum_{t=2}^{T}\sum_{a\in A,b\in B}\left(\vec{X}_t^i(a)-m_i\right)\left(\vec{X}_t^j(b)-m_j\right)$$

where $n_i=|\vec{X}_t^i|$ *and* $\bar{T}=T-1$.

Proof. The principles used in the proof of Proposition 1 are applied because the $\Sigma^{i,i}$ and $\Sigma^{i,j}$ are block matrices. □

Learning Linear Coefficient is to estimate the linear coefficient $\mathbf{B}$ between X_{t-1} and X_t. In this case, given other parameters, Gaussian noise of RTMs and ROMs, the MLE of $\mathbf{B}$ is derived as follows (Roweis and Ghahramani 1999),

$$\widehat{\mathbf{B}}=\left(\sum_{t=2,\cdots,T}X_tX_{t-1}^T\right)\left(\sum_{t=1,\cdots,T-1}X_tX_t^T\right)^{-1}$$

Here, we call the $\widehat{\mathbf{B}}$ linear coefficient of the ground TM, and $\widehat{\mathbf{B}}$ is reshaped as a block matrix by averaging the coefficient in each subblock. When $\widehat{\mathbf{B}}^{i,j}$ denotes the subblock for the linear transition from X_{t-1}^j to X_t^i. The MLE of the block coefficient is represented as follows,

$$\mathbf{B}^{i,i}=\begin{bmatrix}b & b' & \cdots & b'\\ b' & b & \cdots & b'\\ \vdots & \vdots & \ddots & \vdots\\ b' & b' & \cdots & b\end{bmatrix}\mathbf{B}^{i,j}=\begin{bmatrix}b'' & \cdots & b''\\ \vdots & \ddots & \vdots\\ b'' & \cdots & b''\end{bmatrix}$$

such that,

$$b=\tfrac{1}{n_i}\textstyle\sum_{k=1}^{n_i}\widehat{\mathbf{B}}_{k,k}^{i,i},\ b'=\tfrac{1}{n_i(n_i-1)}\sum_{\substack{(k,l)=(1,1)\\k\neq l}}^{(n_i,n_i)}\widehat{\mathbf{B}}_{k,l}^{i,i},$$

$$b''=\tfrac{1}{n_in_j}\textstyle\sum_{(k,l)=(1,1)}^{(n_i,n_j)}\widehat{\mathbf{B}}_{k,l}^{i,j}$$

BlockAverage in Algorithm *LearningRKF* denotes this averaging-out procedure. The block coefficient matrix $\mathbf{B}$ is also the MLE of RTM.

3.3 Learning Observation Models

Given RTMs and a sequence of full observations, we derive the estimate X^t at time step t assuming that there is no observation noise.

Learning Observation Noise is to estimate the mean vector and covariance matrix for the ROM. The MLEs problem is formulated as follows,

$$(\mu_{\text{MLE}},\Sigma_{\text{MLE}})\ =\ \underset{\mu_O,\Sigma_O}{\arg\max}\sum_{t=1}^{T}\log\mathcal{N}(\vec{O}_t;\mu_O,\Sigma_O).$$

where $\vec{O}_t^i=O_t-\mathbf{C}\cdot X_t^i$. The derivation is similar to RTMs. One can substitute $\vec{O}_t^i$ for $\vec{X}_t^i$ in Proposition 2.

[4] One gets the same result when differentiating m, σ^2 and σ' directly from the log-likelihood.

Learning Linear Coefficient C is to calculate the linear coefficient between X_t and O_t.

$$\widehat{\mathbf{C}} = \left(\sum_{t=1,\cdots,T} O_t X_t^T \right) \left(\sum_{t=1,\cdots,T} X_t X_t^T \right)^{-1}$$

Here $\mathbf{C}$ is also calculated from $\widehat{\mathbf{C}}$ by averaging out each subblock as in learning $\mathbf{B}$.

4 LRKF with Regroupings

With the estimated parameters, the RKF predicts the state variables in the relational linear dynamic models. This section presents a new lifted Kalman filtering algorithm, which approximately regroups degenerated relational structures. Existing lifted Kalman filtering algorithms (Choi, Guzman-Rivera, and Amir 2011; Ahmadi, Kersting, and Sanner 2011) suffer degenerations of relational structures when sparse observations are made.[5] Algorithm *LRKF-Regroup* also degenerates the domains of relational atoms by calling *DegenAtom* when state variables are observed in different time, thus different *Obs* (e.g., $Obs_{(i,a)} \neq Obs_{(i,a')}$). Here, $Obs_{(i,a)}$ stores the most recently observed time for a ground substition a in the i-th atom.

To overcome such degeneracy, *LRKF-Regroup* introduces a new subroutine, called *MergeAtom*, which merges covariance structures when random variables are not directly observed for a certain time steps, say k.

Algorithm 2 LRKF-Regroup (Prediction w/ testing data)

```
Input: params $(\mathbf{B}, \Sigma_T, \mathbf{C}, \Sigma_O)$, obs $(O_1, \cdots, O_T)$
repeat
    $\mu_0 \leftarrow 0$, $\Sigma_0 \leftarrow 0$
    $(Obs_{(1,1)}, \cdots, Obs_{(n,n_i)}) \leftarrow (0, \cdots, 0)$
    for $t \leftarrow 1$ to $T$ do
        $(\mu'_t, \Sigma'_t) \leftarrow$ Predict-RTM$(\mu_{t-1}, \Sigma_{t-1}, \mathbf{B}, \Sigma_T)$
        for all $(i, a)$ s.t. $O^i_t(a)$ is observed do
            $Obs_{(i,a)} \leftarrow t$
        end for
        $(\mathbf{B},\Sigma_T,\mathbf{C},\Sigma_O) \leftarrow$ DegenAtom$(Obs, \mathbf{B},\Sigma_T, \mathbf{C}, \Sigma_O)$
        $(\mu_t, \Sigma_t) \leftarrow$ Update-ROM$(\mu'_t, \Sigma'_t, \mathbf{C}, \Sigma_O)$
        $(\mathbf{B},\Sigma_T,\mathbf{C},\Sigma_O) \leftarrow$ MergeAtom$(Obs$, t, $\mathbf{B},\Sigma_T,\mathbf{C},\Sigma_O)$
    end for
until $t$ is $T$
Output: state estimations $((\mu_0, \Sigma_0), \cdots, (\mu_T, \Sigma_T))$
```

The *MergeAtom* operation iterates atoms and find all state variables which are not observed for a certain time steps k. The selected variables are stored in $mlist$. Then, the *BlockMerge* respectively averages diagonal entries and non-diagonal entries in $mlist$, and sets the averaged values to state variables in $mlist$. In this way, it rebuilds the compact relational structure again.

Algorithm 3 MergeAtom

```
input: recent obs time $Obs$, time $t$, params $(\mathbf{B},\Sigma_T,\mathbf{C},\Sigma_O)$
$mlist \leftarrow \emptyset$
for $i = 1$ to $n$ do
    for each $a$ s.t. $Obs_{(i,a)}+k \leq t$ do
        $mlist^i \leftarrow mlist^i \bigcup \{a\}$
    end for
end for
$(\mathbf{B}',\Sigma'_T, \mathbf{C}', \Sigma'_O) =$ BlockAverage$(mlist, \mathbf{B}, \Sigma_T, \mathbf{C}, \Sigma_O)$
output: merged relational structures $(\mathbf{B}', \Sigma'_T, \mathbf{C}', \Sigma'_O)$
```

Lemma 3. *When (1) at least one relational observation is made on O_i at each time step; the variance of any state variable $X^i(a)$ in* LRKF-Regroup *is bounded by ${\sigma'^i_{ROM}}^2$ and converges to $\sqrt{{\sigma'^i_{ROM}}^2 {\sigma^i_{RTM}}^2 + \frac{{\sigma'^i_{ROM}}^4}{4} - \frac{{\sigma^i_{RTM}}^2}{2}}$.*

Proof. Let the variance of the i-th atom be σ^i_{RTM} and the variances of direct and relational observations respectively be ${\sigma^i_{ROM}}^2$ and ${\sigma'^i_{ROM}}^2$ where $\sigma^i_{ROM} < \sigma'^i_{ROM}$ as Equation (4) and (5).

Let $\sigma^i_t(a)^2$ be the variance of a state variable $X^i(a)$ at time t. This variable will be updated by at least one relational observation by the LRKF-Regroup. Then, the new variance is $\frac{1}{\sigma^i_{t+1}(a)^2}=\frac{1}{\sigma^i_t(a)^2} + \frac{1}{{\sigma'^i_{ROM}}^2}$. That is, $\sigma^i_{t+1}(a)^2 \leq \min(\sigma^i_t(a)^2, {\sigma'^i_{ROM}}^2)$. Use the following equation for transition and update in each filtering step,

$$\frac{1}{\sigma^i_{t+1}(a)^2} \leq \frac{1}{\sigma^i_t(a)^2 + {\sigma^i_{RTM}}^2} + \frac{1}{{\sigma'^i_{ROM}}^2}$$

For the convergence, let $\sigma^i_{t+1}(a)=\sigma^i_t(a)=\sigma^i_*(a)$. The variance is $\sigma^i_*(a)^2 = \sqrt{{\sigma'^i_{ROM}}^2 {\sigma^i_{RTM}}^2 + \frac{{\sigma'^i_{ROM}}^4}{4} - \frac{{\sigma^i_{RTM}}^2}{2}}$ □

Theorem 4. *When (1) at least one relational observation is made on O_i at each time step; and (2) no direct observation is made on two state variables $X^i(a)$ and $X^i(a')$ at least for k time steps, the difference of the variances of two state variables $\sigma^i_t(a)$ and $\sigma^i_t(a')$ is bounded by $c^k {\sigma^i_{RTM}}^2 (1+\sigma^i_*(a)^2/{\sigma^i_{ROM}}^2)$ where c is $\sigma^i_*(a)^2/(\sigma^i_*(a)^2+{\sigma'^i_{ROM}}^2)$ and $c \leq 1$.*

Proof. We follow the result of the Lemma 3 and use $\sigma^i_*(a)$. The variance of each time step follows the recursive form,

$$\frac{1}{\sigma^i_{t+1}(a)^2} = \frac{1}{\sigma^i_t(a)^2 + {\sigma^i_{RTM}}^2} + \frac{1}{{\sigma'^i_{ROM}}^2}. \tag{8}$$

An exact (non-recursive) formula for $\sigma^i_{t+1}(a)$ is non trivial. Thus, we introduce another simpler, converging sequence,

$$\bar{\sigma}^i_{t+1}(a)^2 = c(\bar{\sigma}^i_t(a)^2 + {\sigma^i_{RTM}}^2).$$

Since $\sigma^i_t(a)^2 - \bar{\sigma}^i_t(a)^2$ is positive and convex when $c < 1$,

$$0{\leq}\bar{\sigma}^i_t(a){\leq}\sigma^i_t(a){\leq}\sigma^i_*(a),\;\; \sigma^i_*(a)^2{-}\bar{\sigma}^i_t(a)^2 \geq \sigma^i_*(a)^2{-}\sigma^i_t(a)^2$$

[5] Note that, the lifted algorithm in (Choi, Guzman-Rivera, and Amir 2011) only valid when the same number of observations are made at the same time steps.

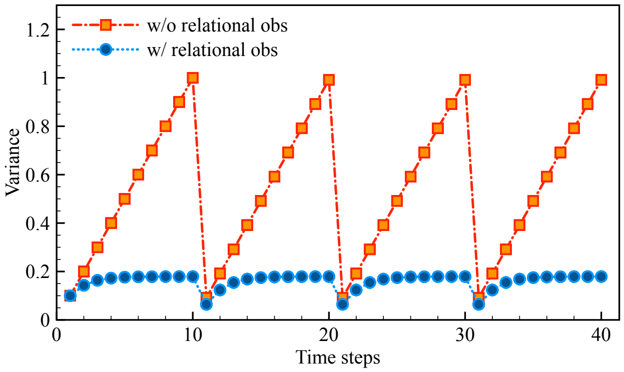

Figure 2: Variances of state variables with one observation per 10 time steps without a relational obs. (square shaped marks) and with relational obs. (circle shaped marks). This figure shows the variances (y axis) in time steps (x axis). Setting that ${\sigma^i_{RTM}}^2$=0.1, ${\sigma^i_{ROM}}^2$=0.1, ${\sigma'^i_{ROM}}^2$=0.5, and let observations made at different time steps. This simulation shows the intuition of Lemma 3.

The convergence of the simpler form is slower than the original one in Equation (8). However, it provides an exact formulation and converge exponentially,

$$\bar{\sigma}^i_t(a)^2 = c^k \bar{\sigma}^i_{t-k}(a)^2 + {\sigma^i_{RTM}}^2 \frac{1-c^k}{1-c}$$

WLOG, we set $X^i(a')$ has no direct observation longer than $X^i(a)$. The variance of $X^i(a')$ has the same formulation with a substitution of $k{+}\alpha$ for k. Thus, variance of $X^i(a')$ is σ^i

$$\bar{\sigma}^i_t(a')^2{=}c^{k+\alpha} \cdot \bar{\sigma}^i_{t-k-\alpha}(a')^2{+}{\sigma^i_{RTM}}^2 \frac{1{-}c^{k+\alpha}}{1{-}c}$$

Note that, $\bar{\sigma}^i_{t+\alpha}(a) \leq \bar{\sigma}^i_{t+\alpha}(a') \leq \bar{\sigma}^i_{t+\infty}(a')$.

$$\begin{aligned}
&|\sigma^i_{t+\alpha}(a')^2{-}\sigma^i_t(a)^2| \leq |\sigma^i_{t+\infty}(a')^2{-}\sigma^i_t(a)^2| \\
&= c^k({\sigma^i_{RTM}}^2/(1-c){-}\sigma^i_{t+k}(a)^2) \\
&= c^k({\sigma^i_{RTM}}^2(1+\sigma^i_*(a)^2/{\sigma^i_{ROM}}^2){-}\sigma^i_{t+k}(a)^2) \\
&\leq c^k {\sigma^i_{RTM}}^2(1+\sigma^i_*(a)^2/{\sigma^i_{ROM}}^2).
\end{aligned}$$

□

5 Experimental Results

To compare the improve the efficiency, we implement the *Lifted RKF* (Choi, Guzman-Rivera, and Amir 2011) and our *LRKF-Regroup*. The algorithm is compared in two datasets with sparse observations: one synthetic dataset and one real-world ground water dataset. Note that, the lifted RKF will not degenerate model on full, dense, observations. In both experiment, we set k to be 4. That is, two state variables will be merged if they have the same observation numbers and types when at least a relational observation is made. With relational observations, the variances of degenerated state variables are reasonably small even after 4 time steps.

In the synthetic dataset, we assume that an atom with 300 ground substitutions, $|X^i|$=30. Then we make a sparse observations with a rate of 90%. That is, 90% of state variables will be observed in each time step. Then, we report the numbers of shattered (or degenerated) groups and average filtering time in the Lifted RKF and in our *LRKF-Regroup*. The result is shown in Figure 3.

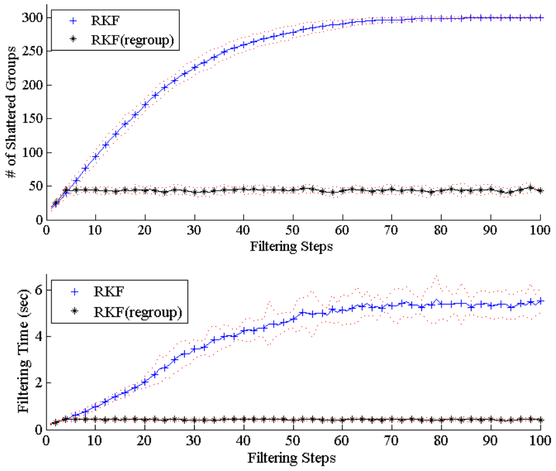

Figure 3: Comparions of Lifted RKF and LRKF-Regoup in a simulation dataset.

To verify the efficiency of Lifted-Regroup, we use a regional groundwater flow MODFLOW model, the Republican River Compact Association (RRCA) Model (McKusick 2003). The extracted dataset is a small subset of the whole ground water model, monthly measured head (water level) at over 3,000 wells. We choose a closely related 200 wells and randomly selecte 100 measurement periods (100 months). Then, we report the degeneracies in *Lifted-RKF* and *LRKF-Regroup* and average filtering time affected by the degenerations in Figure 4.

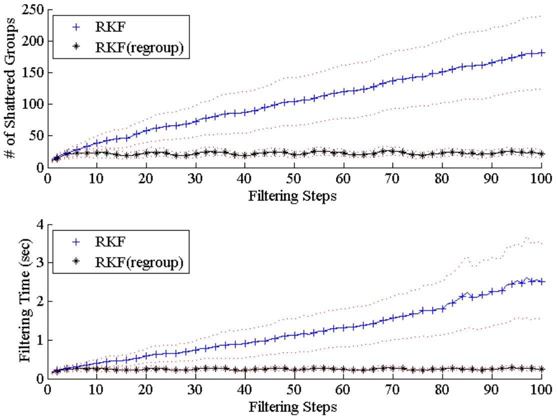

Figure 4: Comparions of Lifted RKF and LRKF-Regoup in the groundwater model.

6 Conclusion

This paper provides new answers and insights on (1) how to learn parameters for RKF; and (2) how to regroup the state variables from noisy real-world data. We propose a new algorithm that regroups the state variables when individual observations are made to RKF in different time steps. We use the RKF in a simulated and a real-world dataset, and demonstrate that the RKF improves the efficiency of filtering the large-scale dynamic system.

7 Acknowledgments

This work is supported by the National Science Foundation Hydrologic Science Program under Grant No. 0943627. This wor was supported by the year of 2014 Research Fund of UNIST (Ulsan National Institute of Science and Technology).

References

Ahmadi, B.; Kersting, K.; and Sanner, S. 2011. Multi-evidence lifted message passing, with application to pagerank and the kalman filter. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

Bahmani-Oskooee, M., and Brown, F. 2004. Kalman filter approach to estimate the demand for international reserves. *Applied Economics* 36(15):1655–1668.

Choi, J.; Guzman-Rivera, A.; and Amir, E. 2011. Lifted relational kalman filtering. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

Choi, J.; Hill, D. J.; and Amir, E. 2010. Lifted inference for relational continuous models. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 126–134.

Clark, M. P.; Rupp, D. E.; Woods, R. A.; Zheng, X.; Ibbitt, R. P.; Slater, A. G.; Schmidt, J.; and Uddstrom, M. J. 2008. Hydrological data assimilation with the ensemble kalman filter: Use of streamflow observations to update states in a distributed hydrological model. *Advances in Water Resources* 31(10):1309 – 1324.

de Salvo Braz, R.; Amir, E.; and Roth, D. 2005. Lifted first-order probabilistic inference. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1319–1325.

Evensen, G. 1994. Sequential data assimilation with a non-linear quasi-geostrophic model using monte carlo methods to forecast error statistics. *Journal of Geophysical Research* 99:10143–10162.

F.P., M., and Bierkens. 2001. Spatio-temporal modelling of the soil water balance using a stochastic model and soil profile descriptions. *Geoderma* 103(1–2):27–50.

Friedman, N.; Getoor, L.; Koller, D.; and Pfeffer, A. 1999. Learning probabilistic relational models. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1300–1309.

Kalman, R. E. 1960. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering* 82(Series D):35–45.

McKusick, V. 2003. Final report for the special master with certificate of adoption of rrca groundwater model. *State of Kansas v. State of Nebraska and State of Colorado, in the Supreme Court of the United States* 3605(1145):4440.

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

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

Roweis, S., and Ghahramani, Z. 1999. A unifying review of linear gaussian models. *Neural Comput.* 11(2):305–345.

Wang, J., and Domingos, P. 2008. Hybrid markov logic networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 1106–1111.