COMP0130: Robotic Vision and Navigation

Lecture 06B: Graphical Models and Factor Graphs

Structure

- Motivation
- Bayesian Filtering
- Graphical Models
- Factor Graphs

What We've Seen So Far…

- In linear systems, the KF produces excellent results but there are issues with computational and storage costs
- In nonlinear systems, the EKF produces poor and divergent results because the dependency structure isn't represented properly
- We need to have better and scalable ways to store the probability

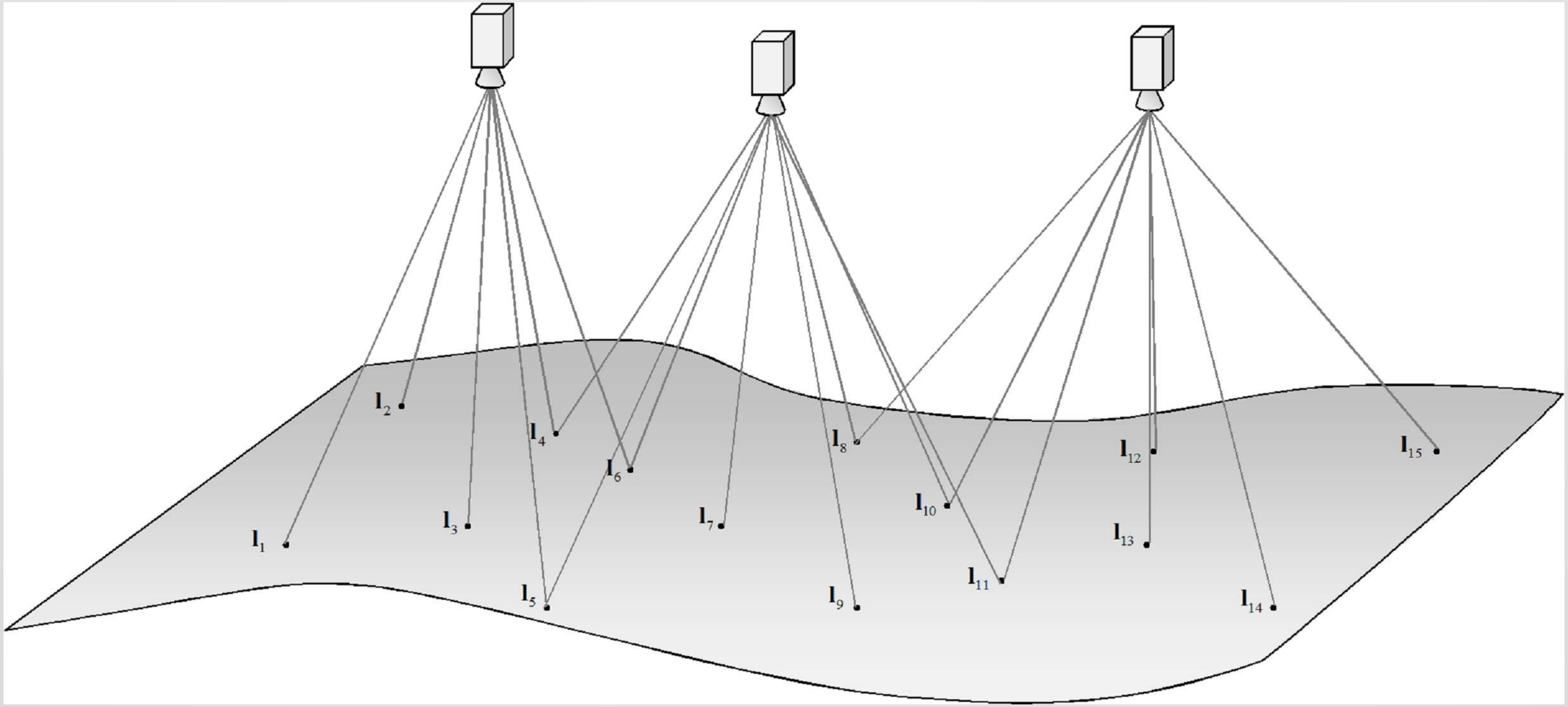
Algorithms for Map Making

- The problem of buildings maps is not restricted to the SLAM community
- In cartography, people need to build maps of the environment

Aerial Photography and Map Making

Part of a photographic plate used for map making from the 1950s

Using Photogrammetry for Map Making

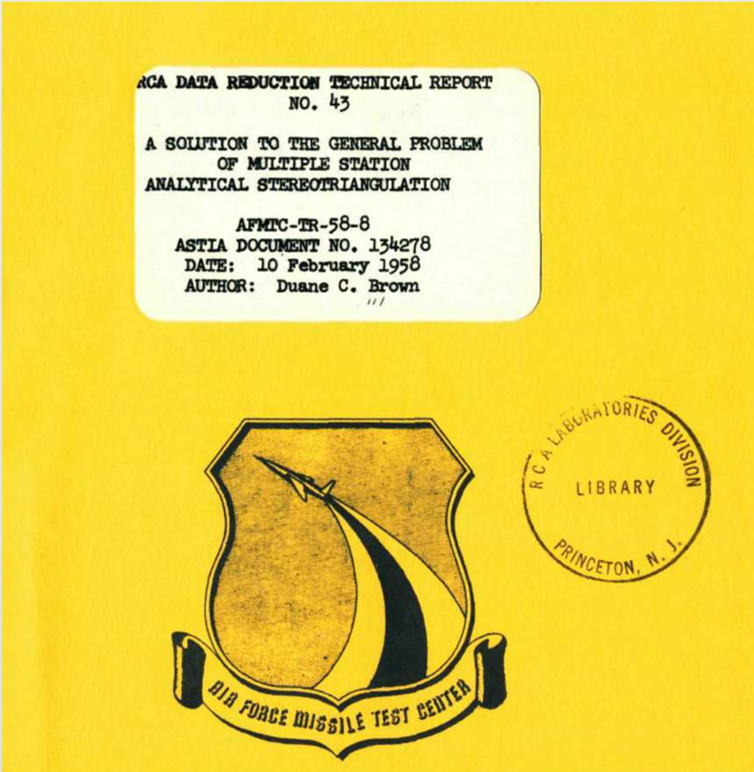


From http://www.geodetic.com/v-stars/what-is-photogrammetry.aspx

Algorithms from Mapping

- The problem of buildings maps is not restricted to the SLAM community
- In cartography, people need to build maps of the environment
- The approach is called *Structure from Motion* and uses *Bundle Adjustment*

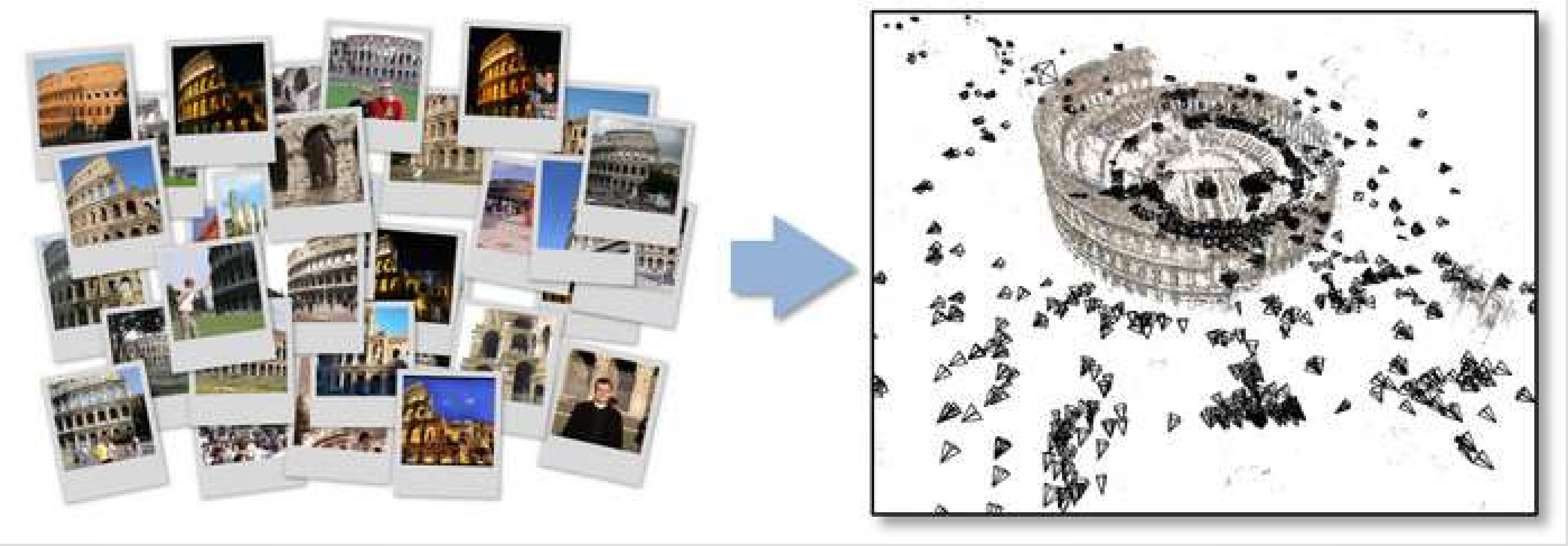
Using Photogrammetry for Map Making


RCA DATA REDUCTION TECHNICAL REPORT
NO. 43

A SOLUTION TO THE GENERAL PROBLEM
OF MULTIPLE STATION
ANALYTICAL STEREOTRIANGULATION

AFMTC-TR-58-8
ASTIA DOCUMENT NO. 134278
DATE: 10 February 1958
AUTHOR: Duane C. Brown

Structure from Motion



Structure from Motion

More Recent Photogrammetry

Photogrammetrically-generated map of Manchester as [a flight sim mod]

SLAM Wants Structure-from-Motion

- Structure-from-motion is an existence proof that it is possible to generate accurate maps of the world
- However, the estimation techniques used are not filtering algorithms
- Rather, they are special cases of an approach called a *Factor Graph*

SLAM Wants Factor Graphs

- Factor Graphs operate in a *completely different way* to Kalman filters
- Therefore, we are going to spend the rest of this lecture actually understanding what these systems are and how to make inferences from them
- Once we've done that, we can return to the SLAM problem again
- We'll start by looking at Bayesian filtering

Bayesian Filtering

- *Motivation*
- Bayesian Filtering
- *Graphical Models*
- *Factor Graphs*

Bayesian Filtering

- The goal is to produce a recursive estimation algorithm a bit like a Kalman filter

- However we will generalise it:
  - We do not restrict ourselves to a linear update rule
  - We propagate the entire probability distribution, not just the first two moments

Bayesian Filtering

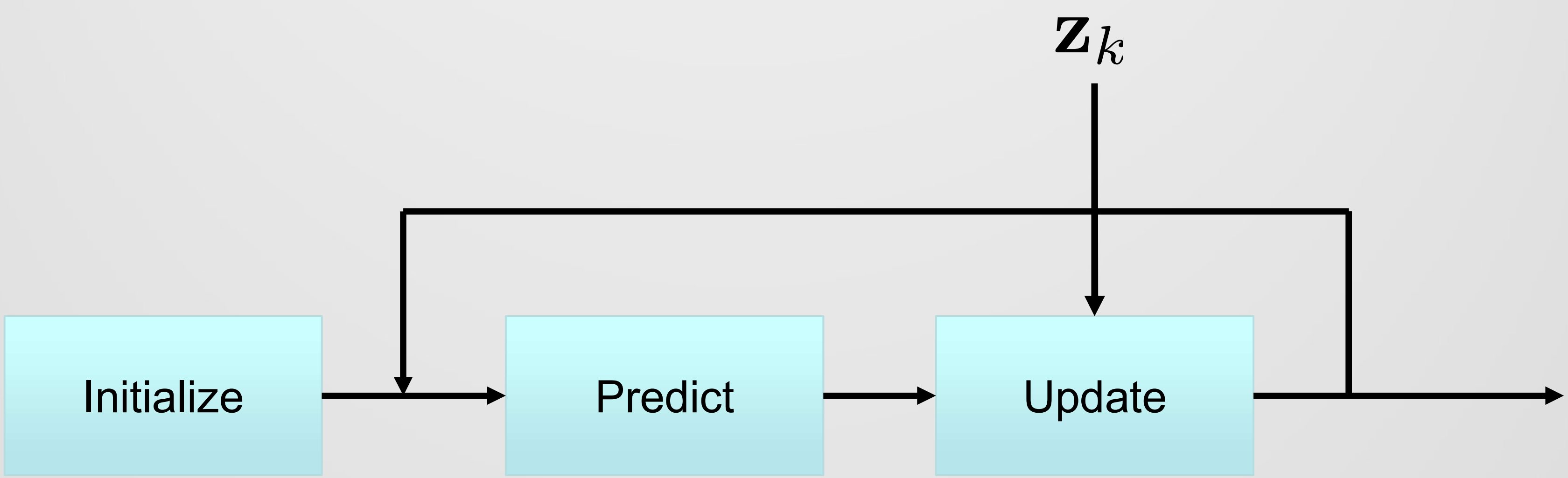
- For the moment we won't look at the SLAM context
- Therefore, the process model is

$$\mathbf{x}_k = \mathbf{f}\left[\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{v}_k\right]$$

- The observation model is

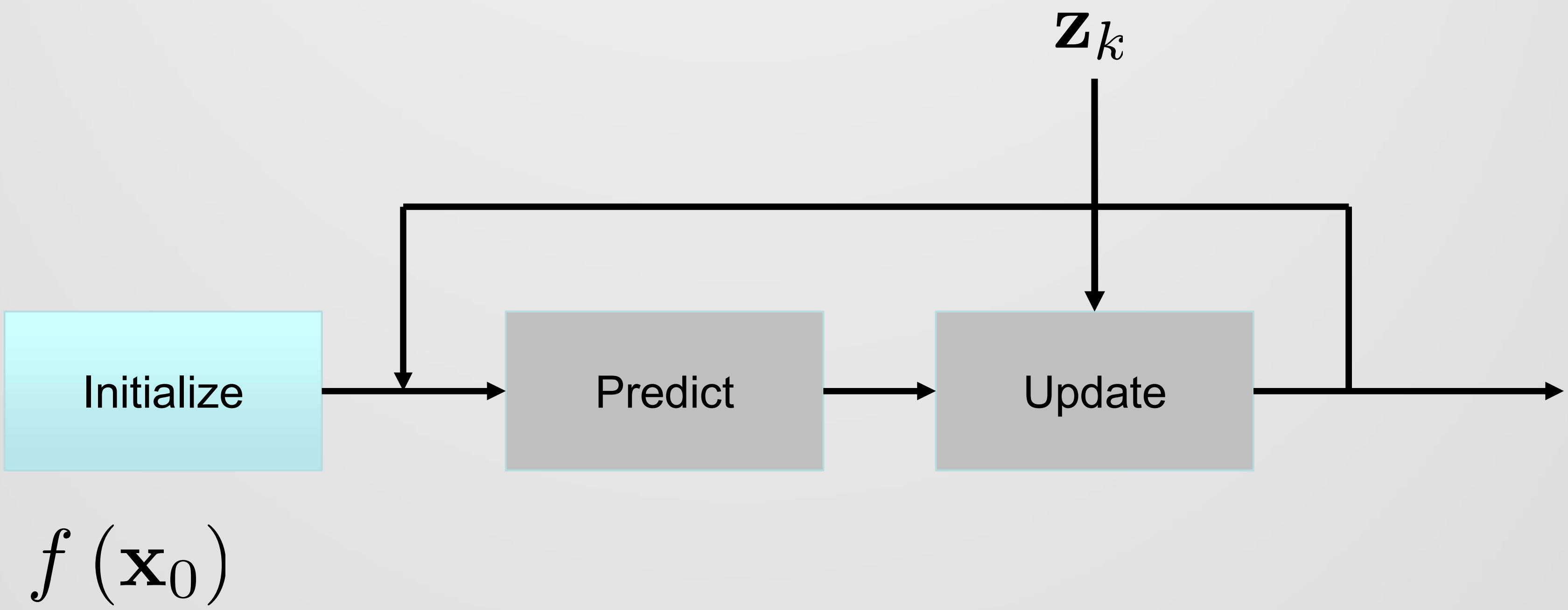
$$\mathbf{z}_k = \mathbf{h}\left[\mathbf{x}_k, \mathbf{w}_k\right]$$

Bayesian Filter

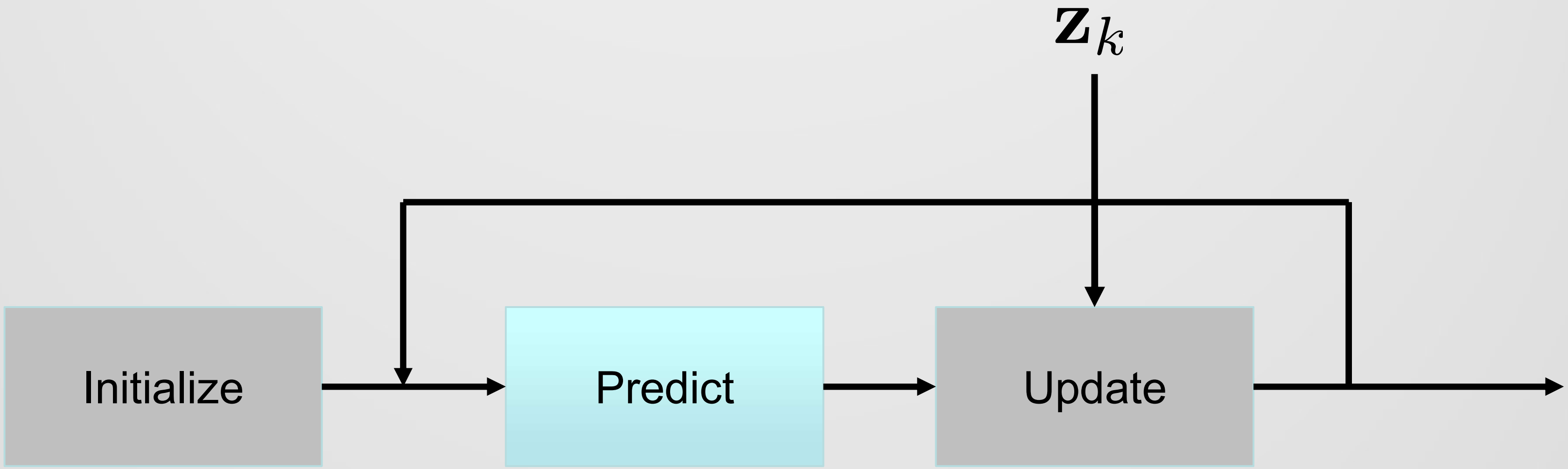


Initialize	Predict	Update
$f\left(\mathbf{x}_0\right)$	$f\left(\mathbf{x}_k \mid \mathbb{I}_{k-1}, \mathbf{u}_k\right)$	$f\left(\mathbf{x}_k \mid \mathbb{I}_k\right)$
$\hat{\mathbf{x}}_{0\mid 0}$	$\hat{\mathbf{x}}_{k\mid k-1}$	$\hat{\mathbf{x}}_{k\mid k}$
$\mathbf{P}_{0\mid 0}$	$\mathbf{P}_{k\mid k-1}$	$\mathbf{P}_{k\mid k}$

Bayesian Filter Initialization



Prediction Step



$$f\left(\mathbf{x}_0\right) \qquad f\left(\mathbf{x}_k \middle| \mathbb{I}_{k-1}, \mathbf{u}_k\right)$$

$$f\left(\mathbf{x}_k \middle| \mathbb{I}_{k-1}, \mathbf{u}_k\right) = f\left(\mathbf{x}_k \middle| \mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k}, \mathbf{x}_0\right)$$

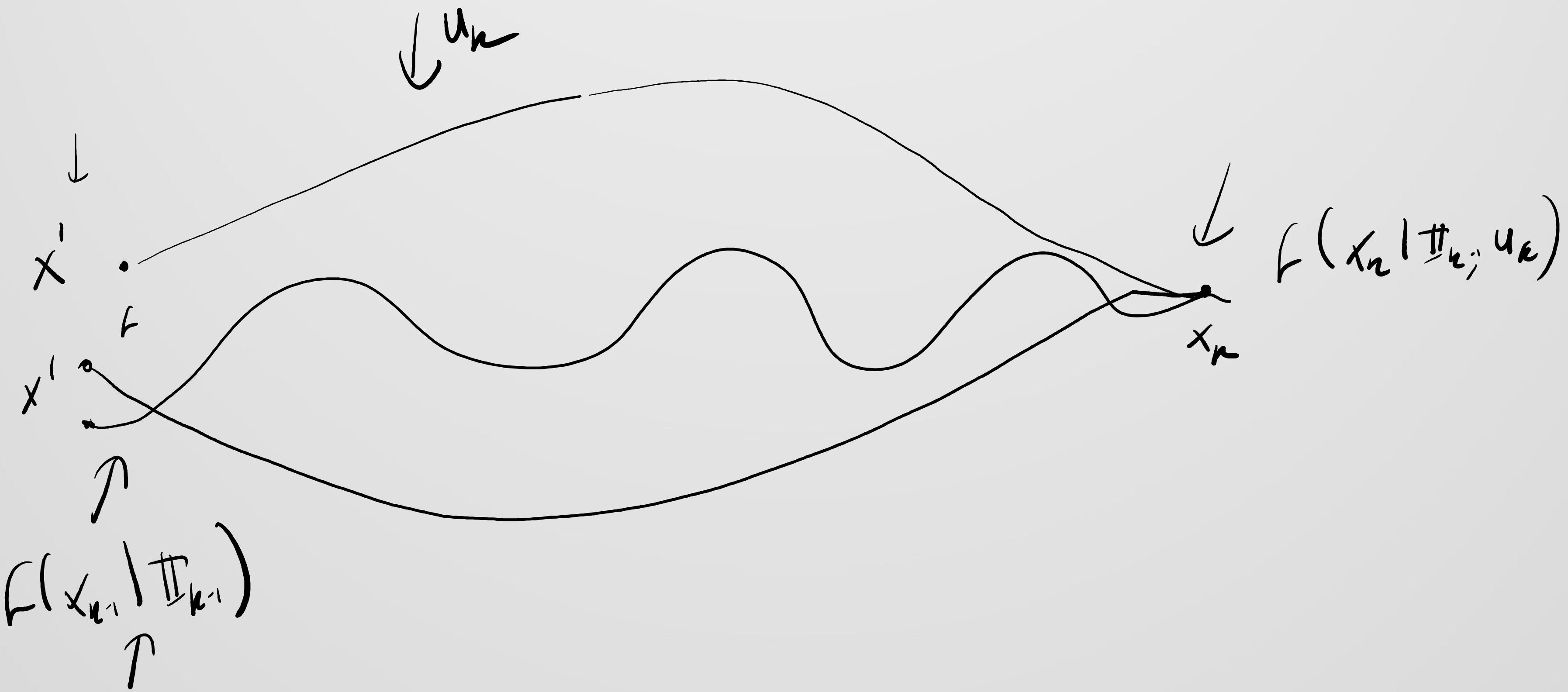
Prediction and Chapman-Kolmogorov

- The predicted distribution is computed from the Chapman-Kolmogorov Equation

$$f\left(\mathbf{x}_k \middle| \mathbb{I}_{k-1}, \mathbf{u}_k\right) = \int_{S} f\left(\mathbf{x}_k \middle| \mathbf{x}', \mathbf{u}_k\right) f\left(\mathbf{x}' \middle| \mathbb{I}_{k-1}\right) \mathrm{d}\mathbf{x}'$$

The Chapman-Kolmogorov Equation

$$f\left(\mathbf{x}_k | \mathbb{I}_{k-1}, \mathbf{u}_k\right) = \int f\left(\mathbf{x}_k | \mathbf{x}', \mathbf{u}_k\right) f\left(\mathbf{x}' | \mathbb{I}_{k-1}\right) \mathrm{d}\mathbf{x}'$$



Working out the State Transition Equation

- We need to have an expression for the state transition probability

$$f\left(\mathbf{x}_k \middle| \mathbf{x}', \mathbf{u}_k\right)$$

- We derive get this from the process model

Understanding Chapman-Kolmogorov

- Recall the process model is

$$\mathbf{x}_k = \mathbf{f}\left[\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{v}_k\right]$$

- Suppose there is an inverse process model which finds all the set of values

$$\mathbf{v}_k = \mathbf{e}\left[\mathbf{x}_k, \mathbf{x}_{k-1}, \mathbf{u}_k\right]$$

Understanding Chapman-Kolmogorov

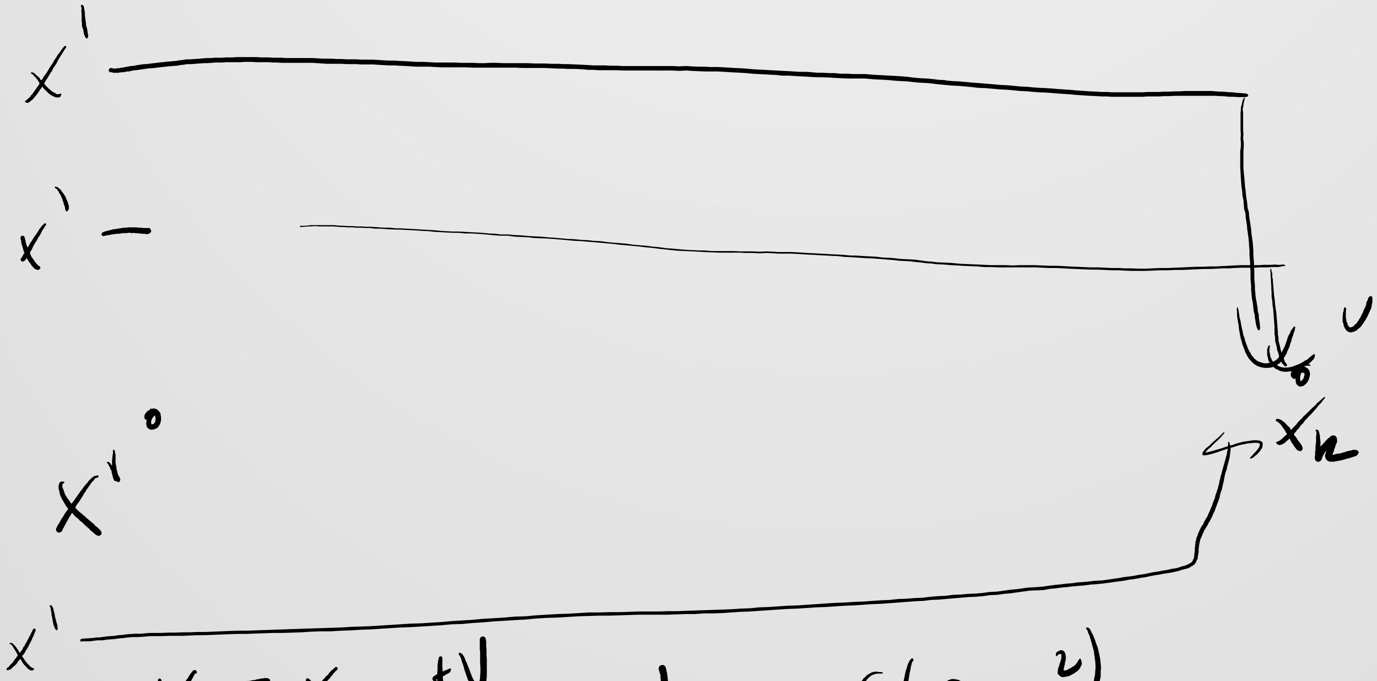
- Therefore,

$$f\left(\mathbf{x}_k \middle| \mathbf{x}', \mathbf{u}_k\right)$$

$$= f_{\mathbf{v}}\left(\mathbf{v}_k = \mathbf{e}\left[\mathbf{x}_k, \mathbf{x}', \mathbf{u}_k\right]\right)$$

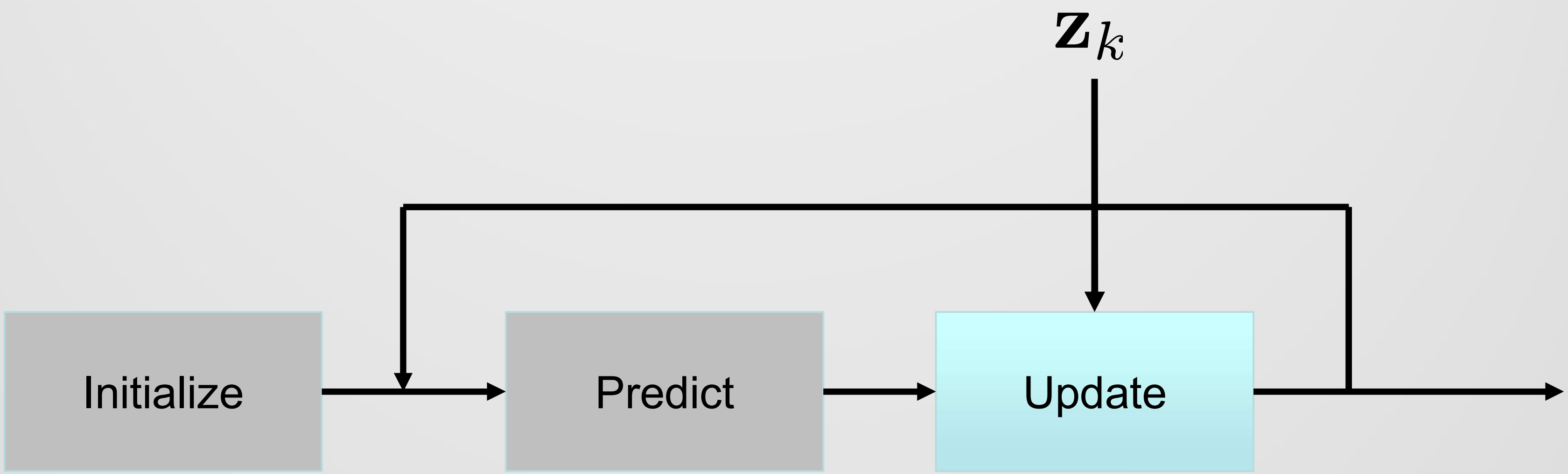
Understanding Chapman-Kolmogorov

$$f\left(\mathbf{x}_k \middle| \mathbf{x}', \mathbf{u}_k\right) = f_{\mathbf{v}}\left(\mathbf{v}_k = \mathbf{e}\left[\mathbf{x}_k, \mathbf{x}', \mathbf{u}_k\right]\right)$$



$$x_k = x_{k-1} + v_k \qquad v_k \sim G(0, \sigma^2)$$

Measurement Update



$$f\left(\mathbf{x}_k \middle| \mathbb{I}_k\right)$$

Measurement Update

- We use Bayes Rule,

$$f\left(\mathbf{x}_k \middle| \mathbb{I}_k\right) = \frac{f\left(\mathbf{z}_k \middle| \mathbf{x}_k\right) f\left(\mathbf{x}_k \middle| \mathbb{I}_{k-1}, \mathbf{u}_k\right)}{f\left(\mathbf{z}_k \middle| \mathbb{I}_{k-1}, \mathbf{u}_k\right)}$$

Measurement Likelihood Equation

- We compute the likelihood from the observation equation

$$\mathbf{z}_k = \mathbf{h}\left[\mathbf{x}_k, \mathbf{w}_k\right]$$

- We assume that we have an inverse observation model of the form

$$\mathbf{w}_k = \mathbf{l}\left[\mathbf{z}_k, \mathbf{x}_k\right]$$

Measurement Likelihood Equation

- Therefore, the measurement likelihood equations,

$$f(\mathbf{z}_k|\mathbf{x}_k) = f_{\mathbf{w}}\left(\mathbf{w}_k = \mathbf{l}\left[\mathbf{x}_k, \mathbf{z}_k\right]\right)$$

- This is sometimes written as the likelihood function

$$f(z_k|x_k) = L(x_k; z_k)$$

$$f\left(\mathbf{z}_k|\mathbf{x}_k\right) = L\left(\mathbf{x}_k; \mathbf{z}_k\right)$$

Normalization Constant

- We also have to compute the normalization constant

$$f\left(\mathbf{z}_k|\mathbb{I}_{k-1},\mathbf{u}_k\right)$$

$$= \int f\left(\mathbf{z}_k|\mathbf{x}'\right) f\left(\mathbf{x}'|\mathbb{I}_{k-1},\mathbf{u}_k\right)\mathrm{d}\mathbf{x}'$$

Bayesian Filters

- Bayesian filters are the optimal solution for filtering and estimation
- They describe exactly the probability distribution in question with no approximation and uncertainty
- Therefore, if we could use them in SLAM, all the issues with drift will go away
- However, they are impossible to implement

Why Are They Impossible?

- We have to compute two integrals:

$$f\left(\mathbf{x}_k|\mathbb{I}_{k-1},\mathbf{u}_k\right)=\int f\left(\mathbf{x}_k|\mathbf{x}',\mathbf{u}_k\right)f\left(\mathbf{x}'|\mathbb{I}_{k-1}\right)\mathrm{d}\mathbf{x}'$$

$$f\left(\mathbf{z}_k|\mathbb{I}_{k-1},\mathbf{u}_k\right)=\int f\left(\mathbf{z}_k|\mathbf{x}'\right)f\left(\mathbf{x}'|\mathbb{I}_{k-1},\mathbf{u}_k\right)\mathrm{d}\mathbf{x}'$$

- In most systems closed form solutions do not exist

Approximate Bayesian Filters

- Approximate solutions have been derived
- These include (retroactively) the Kalman filter and particle filters
- However, all these approaches introduce errors which will accumulate in SLAM

Integration-Free Approaches

- We must develop new methods to do Bayes filtering which avoids the need to integrate in either the prediction or the update steps
- We will use *graphical models* to avoid integration in the prediction step
- We will use *maximum likelihood* estimation to avoid integration in the update step
- For the rest of this lecture we'll look at graphical models and factor graphs

Graphical Models

- *Motivation*
- *Bayesian Filtering*
- Graphical Models
- *Factor Graphs*

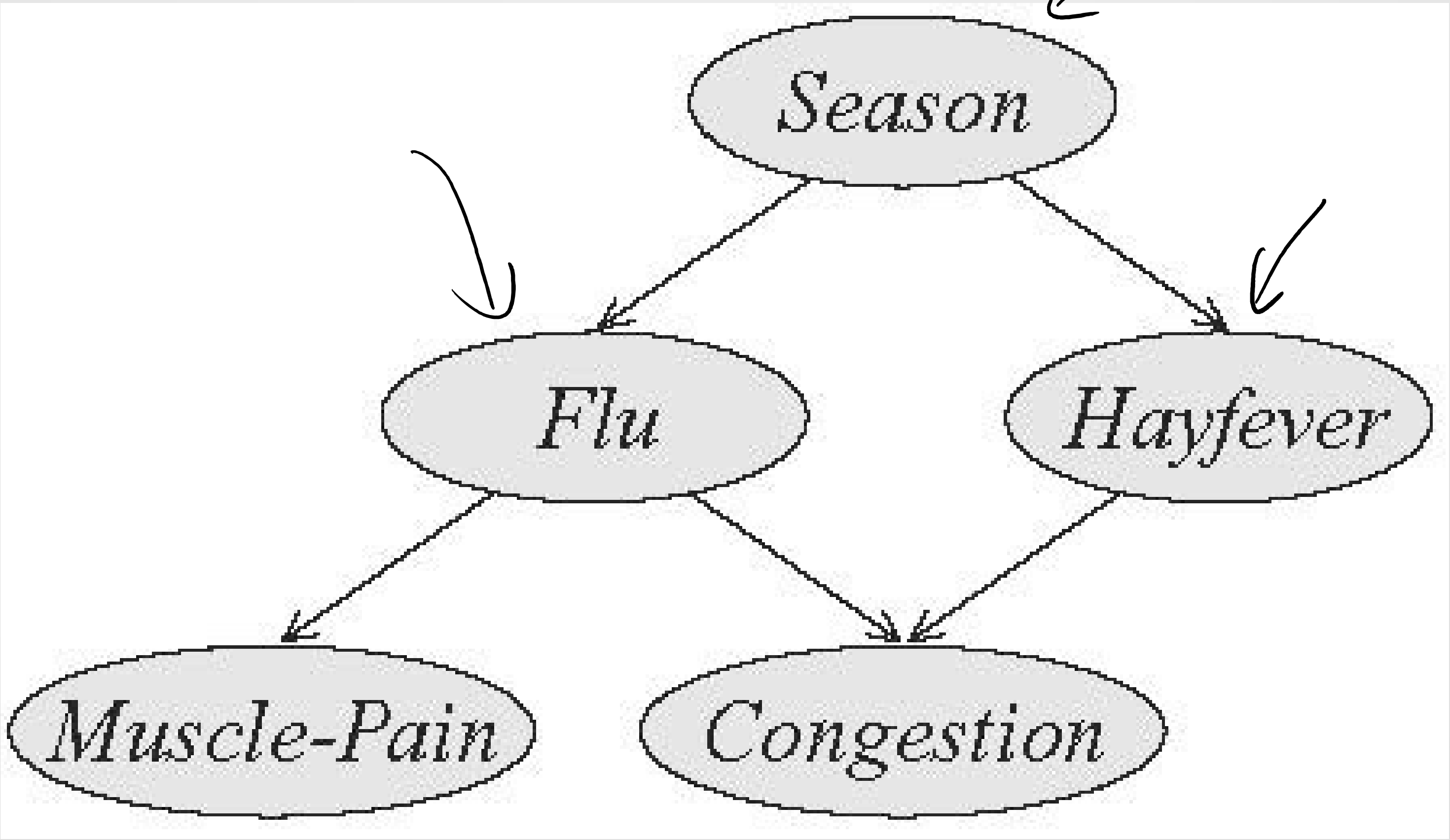
Graphical Models

- Graphical models are a general way to describe probability distributions over multiple random variables
- They decompose a probability distribution into a set of conditional random variables and nodes
- This makes it possible to use a “divide and conquer” strategy to simplify analysis

Graphical Representation

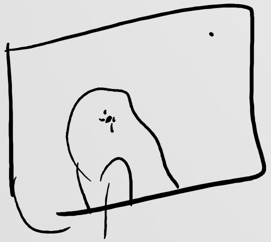
- We can also represent this as a graph:
  - The *vertices* denote the variables or events of interest
  - The *edges* specify conditional probabilities between those variables or events
- This representation is incredibly general

Probabilistic Graph of Disease

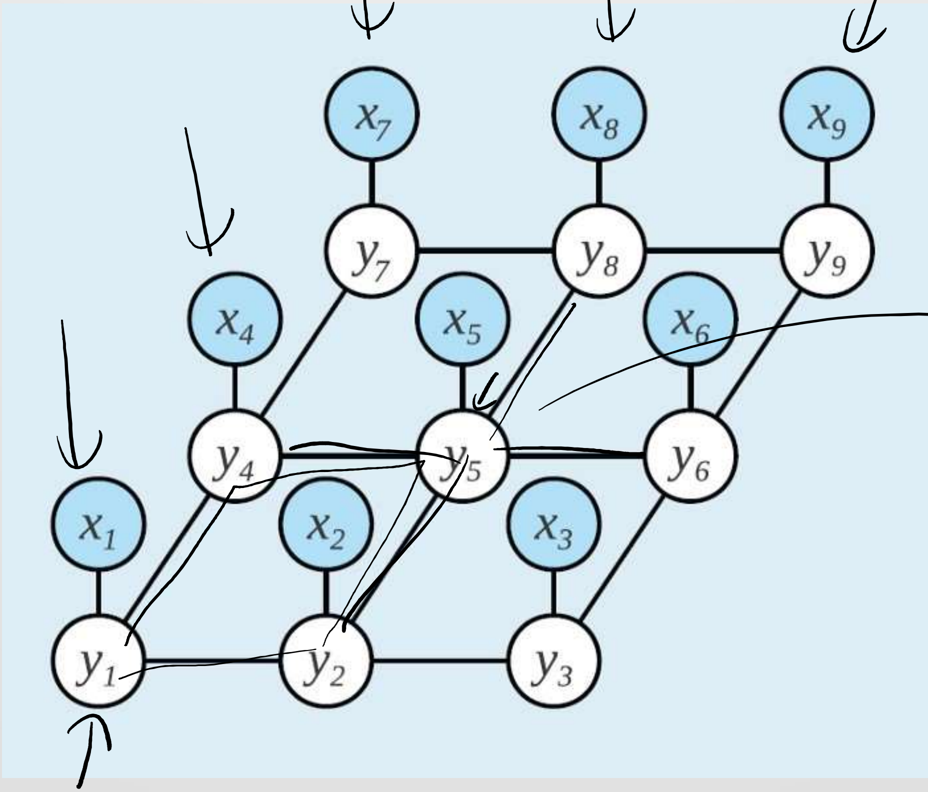


From “Counting non-redundant parameters in graphical models”

Probabilistic Graph of Pixel Labels



tree



y_{200}

tree

From “Understanding Graphical Models Intuitively”

Probabilistic Graphical Models

- Consider the joint probability over two random variables

$$f(\mathbf{a}, \mathbf{b})$$

- From conditional probability, we can write this as:

$$f(\mathbf{a}, \mathbf{b}) = f(\mathbf{a}) f(\mathbf{b}|\mathbf{a})$$

Graphical Representation

$$f(\mathbf{a}, \mathbf{b}) = f(\mathbf{a}) f(\mathbf{b}|\mathbf{a})$$

Graphical Representation

- Suppose we now consider the joint distribution over five random variables

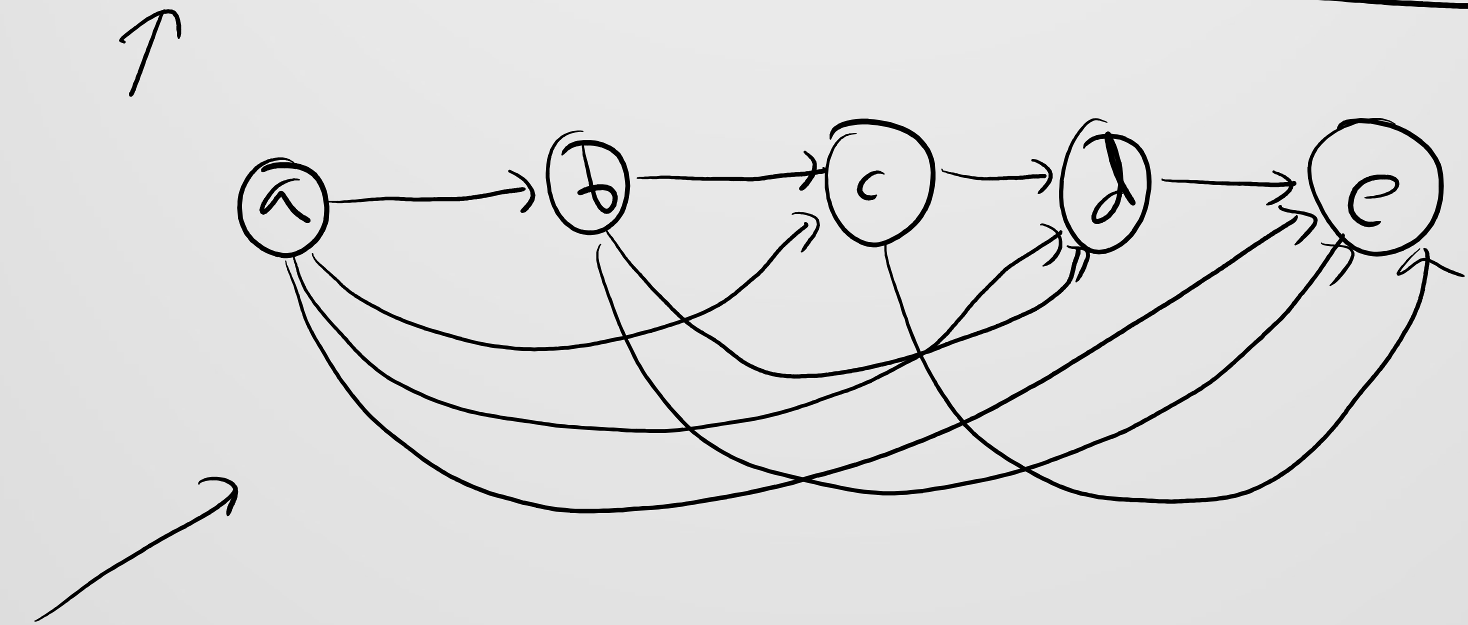
$$f(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e})$$

- Using conditional probabilities, this is

$$\begin{aligned} f(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}) = {} & f(\mathbf{a}) f(\mathbf{b}|\mathbf{a}) f(\mathbf{c}|\mathbf{a}, \mathbf{b}) \\ & \times f(\mathbf{d}|\mathbf{a}, \mathbf{b}, \mathbf{c}) f(\mathbf{e}|\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}) \end{aligned}$$

Graphical Model Representation

$$f(\mathbf{a})f(\mathbf{b}|\mathbf{a})f(\mathbf{c}|\mathbf{a},\mathbf{b})f(\mathbf{d}|\mathbf{a},\mathbf{b},\mathbf{c})f(\mathbf{e}|\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d})$$



Graphical Representation

- If the relationship between all the variables is arbitrary, graphical models don't help very much
- However, in many situations, the graphical models come from actual physical systems
- As a result, there can be a very strong dependency between the random variables
- This applies a very strong structure to the graph

UCL

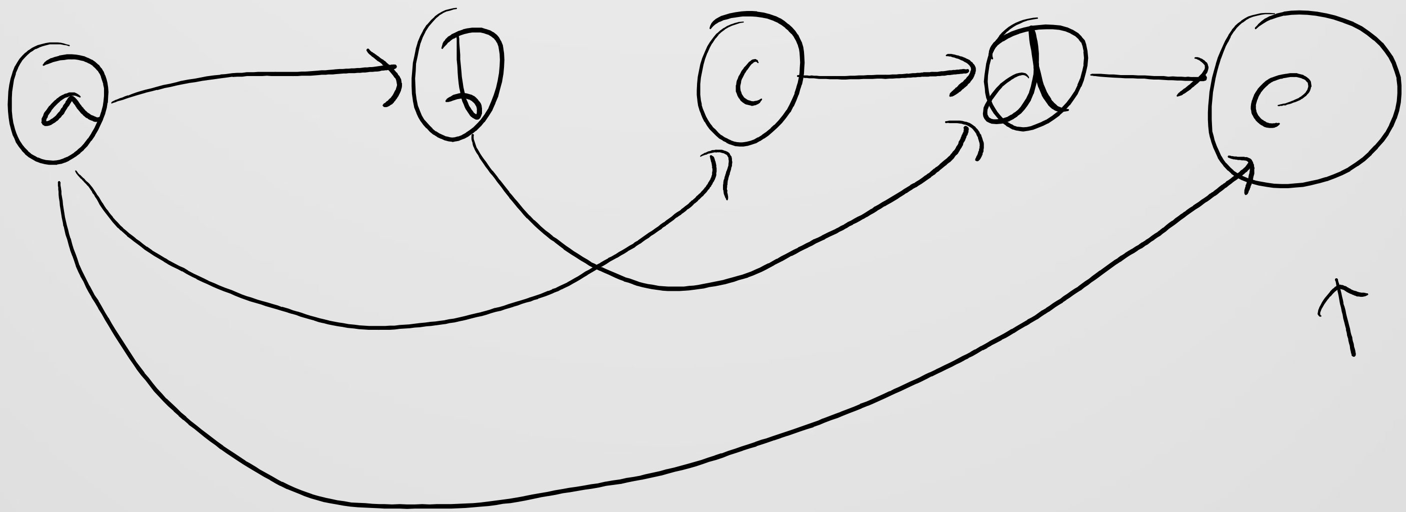
Graphical Representation

- Suppose it turns out that the relationship between the variables is

$$f(\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d},\mathbf{e}) = f(\mathbf{a})f(\mathbf{b}|\mathbf{a})f(\mathbf{c}|\mathbf{a}) \\ \times f(\mathbf{d}|\mathbf{b},\mathbf{c})f(\mathbf{e}|\mathbf{a},\mathbf{d})$$

Graphical Model Representation

$$f(\mathbf{a})f(\mathbf{b}|\mathbf{a})f(\mathbf{c}|\mathbf{a})f(\mathbf{d}|\mathbf{b},\mathbf{c})f(\mathbf{e}|\mathbf{a},\mathbf{d})$$



Graphical Representation of Prediction

- We are now going to look at how to express the prediction problem graphically
- The standard prediction problem is to compute the distribution

$$f\left(\mathbf{x}_k | \mathbb{U}_k\right) \qquad \mathbb{U}_k = \{\mathbf{U}_{1:k}, \mathbf{x}_0\}$$

Bayesian Filter Approach

- The standard Bayesian filter approach is to keep running the prediction step over and over again

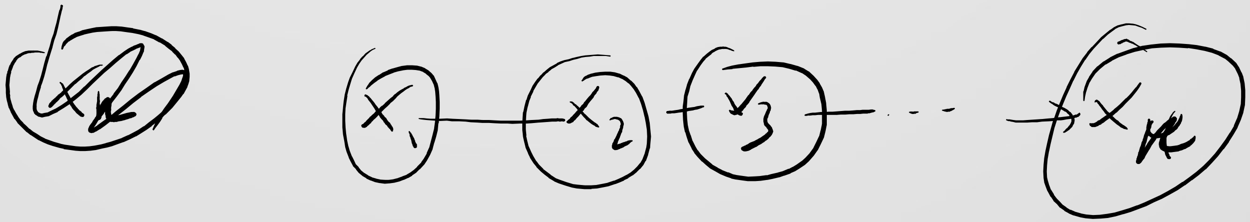
$$
\begin{aligned}
f\left(\mathbf{x}_k \middle| \mathbb{U}_k\right) &= f\left(\mathbf{x}_k \middle| \mathbb{U}_{k-1}, \mathbf{u}_k\right) \\
&= \int \underline{f\left(\mathbf{x}_k \middle| \mathbf{x}', \mathbf{u}_k\right) f\left(\mathbf{x}' \middle| \mathbb{U}_{k-1}\right) \mathrm{d}\mathbf{x}'}
\end{aligned}
$$

- However, this introduces the integration issues

Graphical Model Approach to Prediction

- Suppose, instead, that we predict the *entire history* of the state of the platform,

$$f\left(\mathbf{x}_{1:k} \middle| \mathbb{U}_k\right)$$



Joint History State Space

- The space consists of the estimate of the entire state history of the platform over time,

x_k

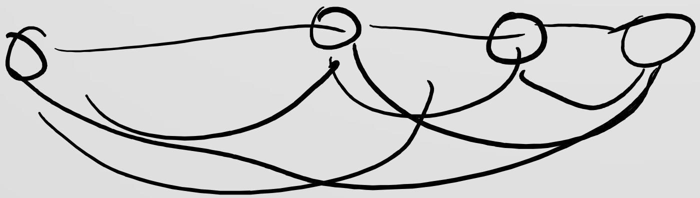
$$\mathbf{x}_{1:k} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_k \end{bmatrix}$$

x_0

Graphical Model Approach to Prediction

- Although this model contains more variables than the filtering case, it's actually easier to compute

- The reason is that the process model introduces a very strong constraint on the temporal relationship between variables



Predicting the First Step

- The joint prediction is given by

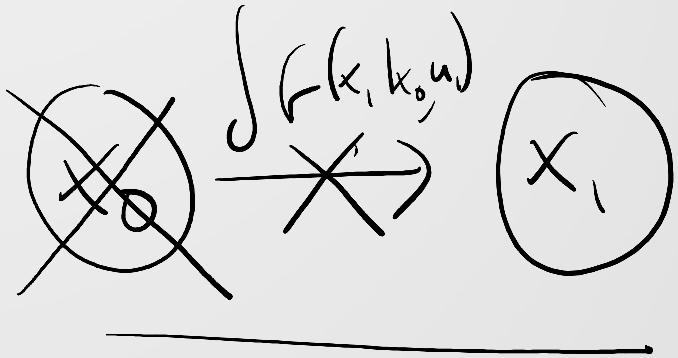
$$f\left(\mathbf{x}_1 \middle| \mathbf{u}_1, \mathbf{x}_0\right) = f\left(\mathbf{x}_1 \middle| \mathbf{x}_0, \mathbf{u}_1\right) f\left(\mathbf{x}_0\right)$$

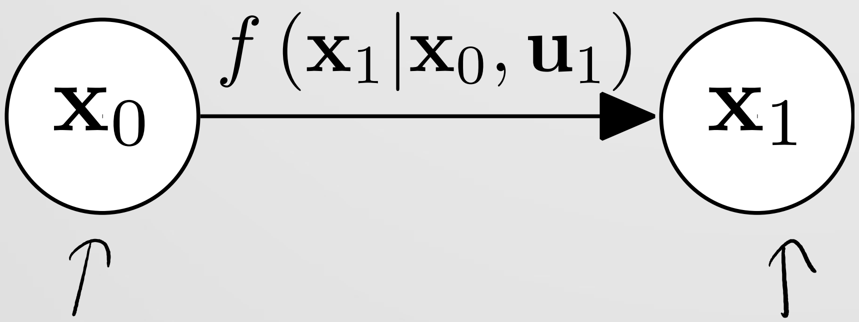
State transition density

Prior

$p(x_1 | U_1)$

Graph for the First Step





Predicting the Second Step

- Now suppose we want to compute

$$f\left(\mathbf{x}_{1:2} \middle| \mathbb{U}_2\right)$$

- We can follow the same decomposition strategy and write

$U_2 = \{u_1, u_2\}$

$$f\left(\mathbf{x}_{1:2} \middle| \mathbb{U}_2\right) = f\left(\mathbf{x}_2 \middle| \mathbf{x}_1, \mathbb{U}_2\right) f\left(\mathbf{x}_1 \middle| \mathbb{U}_2\right)$$

Graphical Model Approach to Prediction

- Recall from the process model that

$$\mathbf{x}_k = \mathbf{f}\left[\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{v}_k\right]$$

- Therefore, the state at time k just depends on the state at time $k-1$ and the control input at k
- It does not depend on the state or control at any other timestep

Prediction

- Therefore, the equations simplify to

$\leftarrow u_1, \cancel{u_2}, x_0$

$$
\begin{cases}
f\left(\mathbf{x}_1 \middle| \mathbb{U}_2\right) = f\left(\mathbf{x}_1 \middle| \mathbf{x}_0, \mathbf{u}_1\right) \leftarrow \\
f\left(\mathbf{x}_2 \middle| \mathbf{x}_1, \mathbb{U}_2\right) = \underline{f\left(\mathbf{x}_2 \middle| \mathbf{x}_1, \mathbf{u}_2\right)} \leftarrow
\end{cases}
$$

$\cancel{u_1}, u_2, \cancel{x_0}$

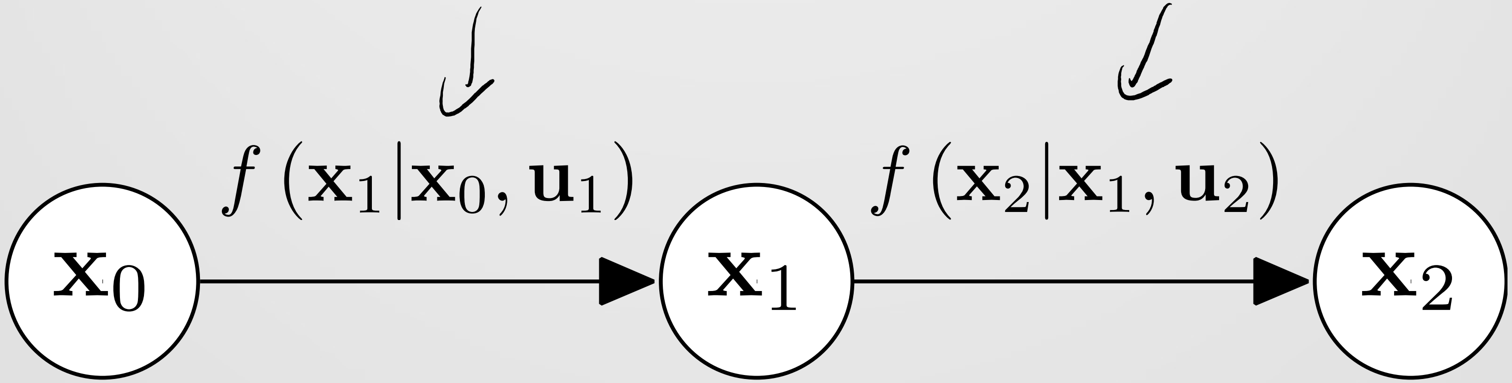
Prediction

- Substituting, we get

$$f\left(\mathbf{x}_{1:2} \middle| \mathbb{U}_2\right) \overset{=}{\cancel{\propto}} f\left(\mathbf{x}_2 \middle| \mathbf{x}_1, \mathbf{u}_2\right) f\left(\mathbf{x}_1 \middle| \mathbf{x}_0, \mathbf{u}_1\right) \times f\left(\mathbf{x}_0\right)$$

- This has no integration again

Graph for the Second Step



Prediction

- By induction we can expand this to k timesteps to get

$$f\left(\mathbf{x}_{1:k} | \mathbb{U}_k\right) = \underline{f\left(\mathbf{x}_0\right)} \prod_{i=1}^{k} f\left(\underline{\mathbf{x}_i | \mathbf{x}_{i-1}, \mathbf{u}_i}\right)$$

Graphical Model Approach to Updates

- We will now extend this to include observations
- We now seek to compute

$$f\left(\mathbf{x}_{1:k} \middle| \mathbb{I}_k\right)$$

$$\mathbb{I}_k = \{\mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0\}$$

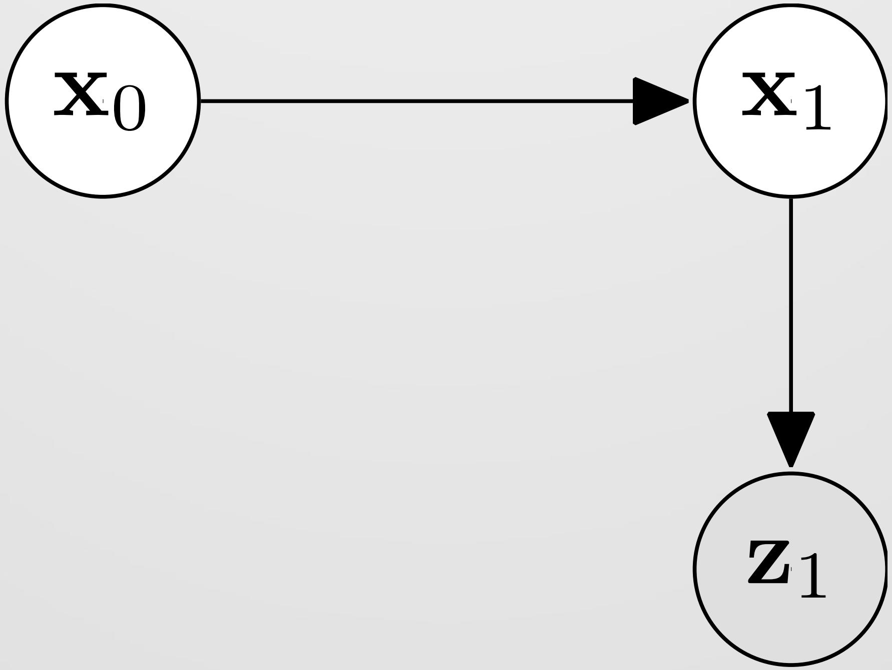
Graphical Model Approach to Updates

- Consider the case when there is a single prediction step followed by a single update

- The joint density in this case is

$$f\left(\mathbf{x}_1 \middle| \mathbb{I}_1\right)$$

In Graphical Form



Incorporating Observations

- We can use Bayes Rule to incorporate the observation,

$$f\left(\mathbf{x}_1 \middle| \mathbb{I}_1\right) = \frac{f\left(\mathbf{z}_1 \middle| \mathbf{x}_1\right) f\left(\mathbf{x}_1 \middle| \mathbb{U}_1\right)}{f\left(\mathbf{z}_1 \middle| \mathbb{U}_1\right)}$$

Incorporating Observations

- Substituting for the predictions,

$$f\left(\mathbf{x}_1 \middle| \mathbb{I}_1\right) = \frac{f\left(\mathbf{z}_1 \middle| \mathbf{x}_1\right) f\left(\mathbf{x}_1 \middle| \mathbf{x}_0, \mathbf{u}_1\right) f\left(\mathbf{x}_0\right)}{f\left(\mathbf{z}_1 \middle| \mathbb{U}_1\right)}$$

Normalization Constant

- We have to compute this from

$$f\left(\mathbf{z}_1|\mathbb{U}_1\right) = \int f\left(\mathbf{z}_1|\mathbf{x}_1'\right) f\left(\mathbf{x}_1'|\mathbb{U}_1\right)\mathrm{d}\mathbf{x}_{0:1}'$$

Constant

$$= \int f\left(\mathbf{z}_1|\mathbf{x}_1'\right) f\left(\mathbf{x}_1'|\mathbf{x}_0', \mathbf{u}_1\right) f\left(\mathbf{x}_0'\right)\mathrm{d}\mathbf{x}_{0:1}'$$

- We will treat it as an (unknown) constant for now
- When we look at inference, we'll see how to eliminate it

Putting it All Together

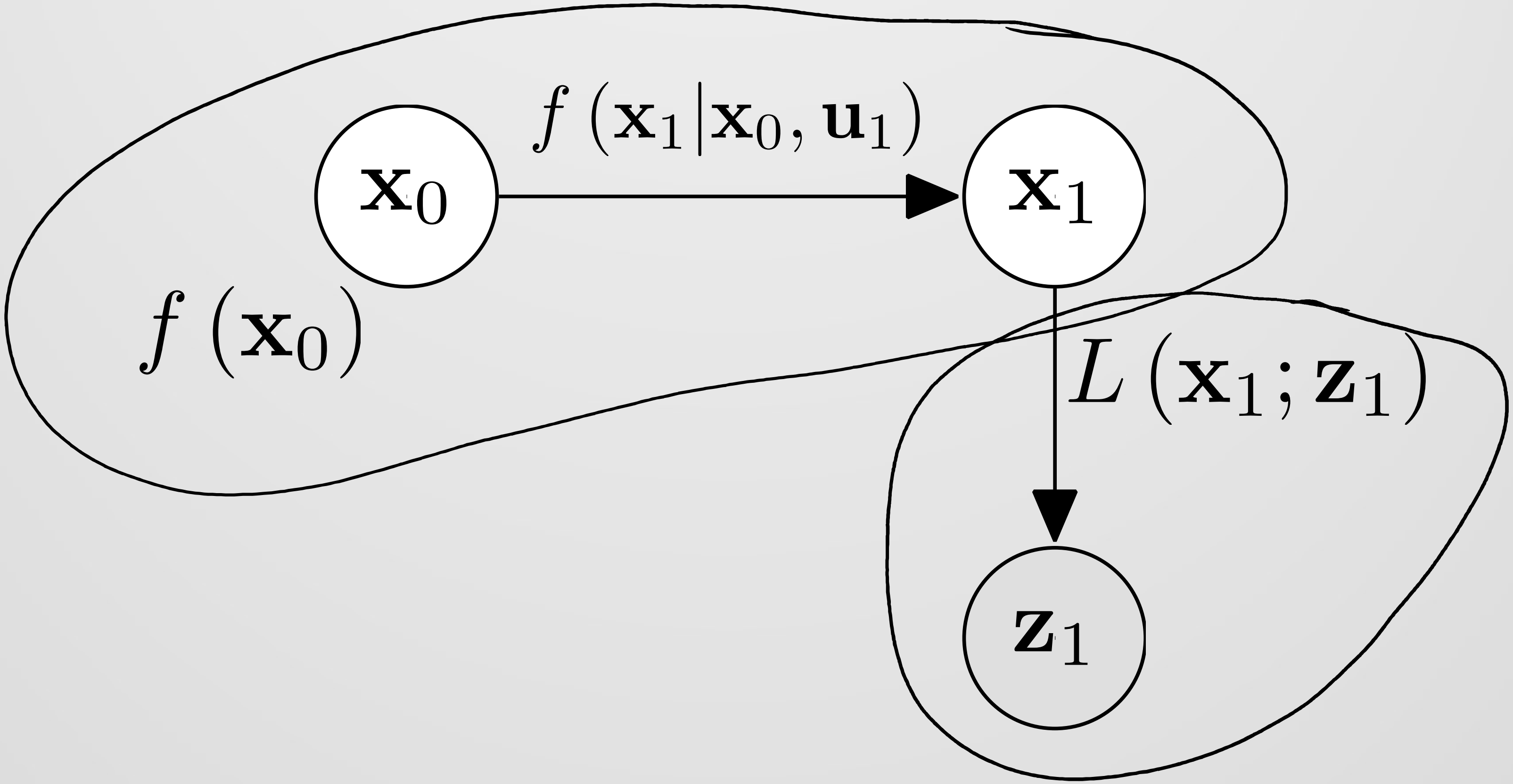
- Substituting everything, the joint density can be written as

$$f\left(\mathbf{x}_1 \middle| \mathbb{I}_1\right) \propto \underline{f\left(\mathbf{z}_1 \middle| \mathbf{x}_1\right)}\ \underline{f\left(\mathbf{x}_1 \middle| \mathbf{x}_0, \mathbf{u}_1\right) f\left(\mathbf{x}_0\right)}$$

- Substituting for the likelihood, we have

$$f\left(\mathbf{x}_1 \middle| \mathbb{I}_1\right) \propto \underline{L\left(\mathbf{x}_1 ; \mathbf{z}_1\right)}\ f\left(\mathbf{x}_1 \middle| \mathbf{x}_0, \mathbf{u}_1\right) f\left(\mathbf{x}_0\right)$$

In Graphical Form



2 Timestep Case

- The form for the 2 timestep case is

$$f\left(\mathbf{x}_{1:2} \middle| \mathbb{I}_2\right) = \frac{f\left(\mathbf{z}_{1:2} \middle| \mathbf{x}_{1:2}\right) f\left(\mathbf{x}_{1:2} \middle| \mathbb{U}_2\right)}{f\left(\mathbf{z}_{1:2} \middle| \mathbb{U}_2\right)}$$

- We'll need to separate out the likelihood and the prediction terms

Likelihood

- Recall from the observation model that the observation at timestep k only depends on the state at timestep k

$$\mathbf{z}_k = \mathbf{h}\left[\mathbf{x}_k, \mathbf{w}_k\right]$$

- Therefore,

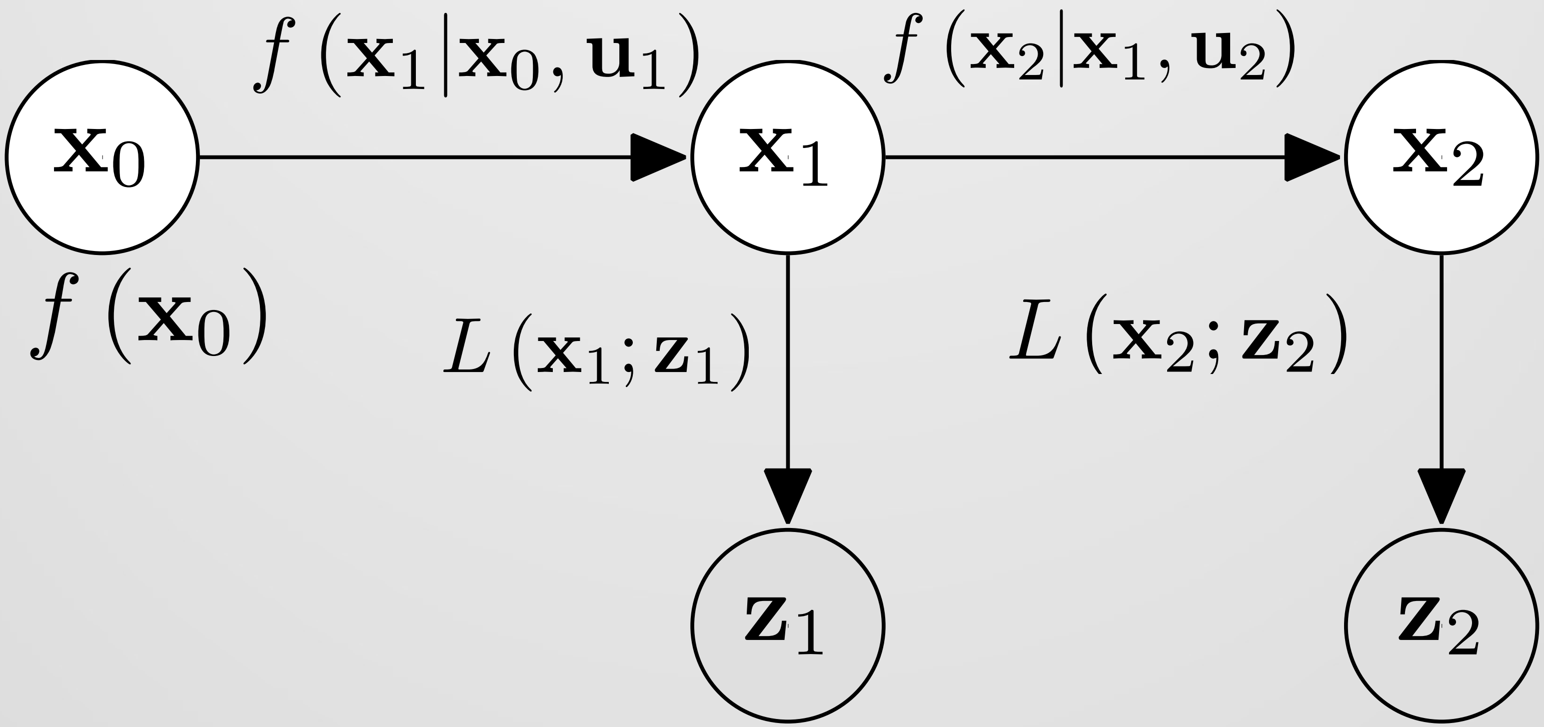
$$\begin{aligned} f\left(\mathbf{z}_{1:2} \middle| \mathbf{x}_{1:2}\right) &= \underline{f\left(\mathbf{z}_2 \middle| \mathbf{x}_2\right)}\, \underline{f\left(\mathbf{z}_1 \middle| \mathbf{x}_1\right)} \\ &= \underline{L(\mathbf{x}_2; \mathbf{z}_2)}\, \underline{L(\mathbf{x}_1; \mathbf{z}_1)} \end{aligned}$$

Incorporating Observations

- Substituting this and the prediction we saw earlier,

$$f\left(\mathbf{x}_{1:2}|\mathbb{I}_2\right) \propto \underline{f\left(\mathbf{x}_2|\mathbf{x}_1,\mathbf{u}_2\right) f\left(\mathbf{x}_1|\mathbf{x}_0,\mathbf{u}_1\right) f\left(\mathbf{x}_0\right)}$$
$$\times \underline{L\left(\mathbf{x}_2;\mathbf{z}_2\right) L\left(\mathbf{x}_1;\mathbf{z}_1\right)}$$

2D Step Case


$f(\mathbf{x}_1|\mathbf{x}_0, \mathbf{u}_1)$
$f(\mathbf{x}_2|\mathbf{x}_1, \mathbf{u}_2)$
$\mathbf{x}_0$
$\mathbf{x}_1$
$\mathbf{x}_2$
$f(\mathbf{x}_0)$
$L(\mathbf{x}_1; \mathbf{z}_1)$
$L(\mathbf{x}_2; \mathbf{z}_2)$
$\mathbf{z}_1$
$\mathbf{z}_2$

General Form

- We can apply this recursively for k timesteps,

f_v

$$f\left(\mathbf{x}_{0:k} \middle| \mathbb{I}_k\right) \propto f\left(\mathbf{x}_0\right) \prod_{i=1}^{k} \underbrace{f\left(\mathbf{x}_i \middle| \mathbf{x}_{i-1}, \mathbf{u}_i\right)}$$

$$\times \prod_{i=1}^{k} \underbrace{L\left(\mathbf{x}_i ; \mathbf{z}_i\right)}_{f_w}$$

Extensions

- The graph is incredibly easy to extend
- For example, if the observations are only intermittently available in the set Z

$$f\left(\mathbf{x}_{0:k} \middle| \mathbb{I}_k\right) \propto f\left(\mathbf{x}_0\right) \prod_{i=1}^{k} f\left(\mathbf{x}_i \middle| \mathbf{x}_{i-1}, \mathbf{u}_i\right)$$

$$\times \prod_{i \in Z} L\left(\mathbf{x}_i ; \mathbf{z}_i\right)$$

Factor Graphs

- *Motivation*
- *Bayesian Filtering*
- *Graphical Models*
- Factor Graphs

Factor Graphs

- The graphical models introduced are examples of Bayes networks
- These networks consist of multiplying terms together
- These can be equivalently represented using factor graphs
- Since factor graphs are extensively used in the literature, we'll define them here

Factor Graphs

- Suppose we want to evaluate the function

$$\underline{g(\mathbf{Y})}, \quad \underline{\mathbf{Y} = \{\mathbf{y}_0, \ldots, \mathbf{y}_n\}}$$

- If this function were arbitrary, there's not a lot we can do
- However, factor graphs assume it can be written as

$$\underline{g(\mathbf{Y})} = \prod_{j=1}^{m} g_j(\mathbf{S}_j), \quad \underline{\mathbf{S}_j \subseteq \mathbf{Y}}$$

General Factor Graph Example

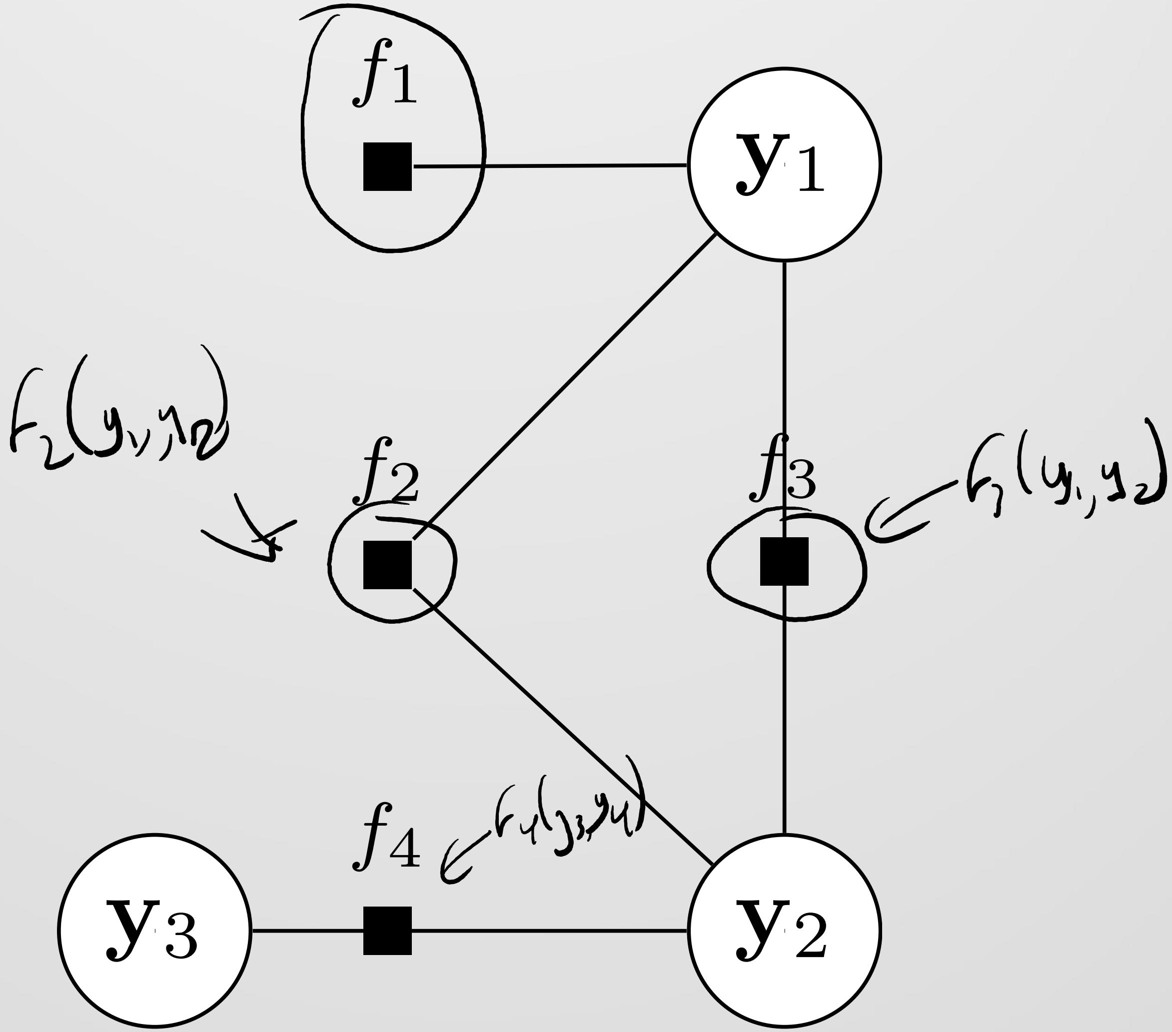
- For example, consider the function

$$g\left(\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3\right)$$

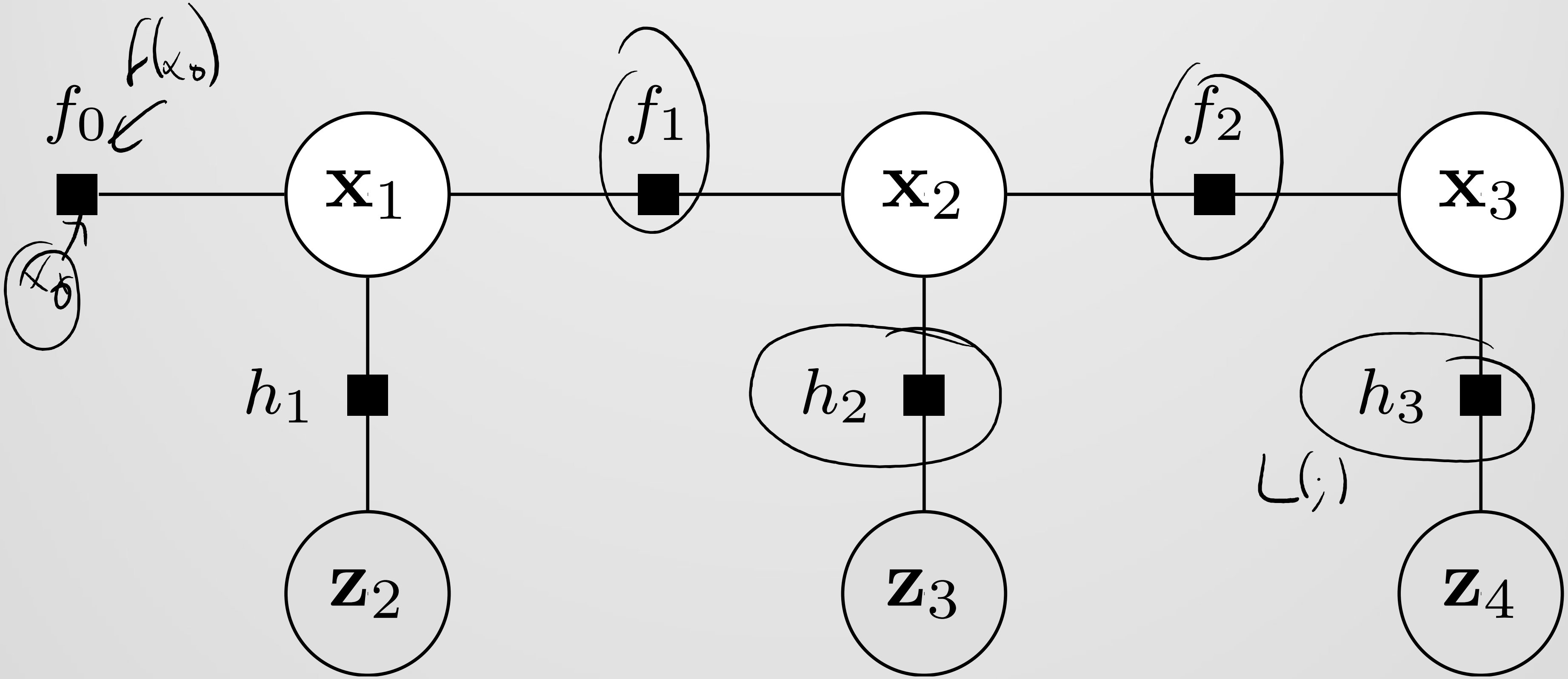
- Suppose it can be factorised as

$$g\left(\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3\right) = f_1\left(\mathbf{y}_1\right) f_2\left(\mathbf{y}_1, \mathbf{y}_2\right) \times f_3\left(\mathbf{y}_1, \mathbf{y}_2\right) f_4\left(\mathbf{y}_2, \mathbf{y}_3\right)$$

Factor Graph Structure



Estimation in a Factor Graph



Summary

- We have introduced graphical models as a way to simplify implementation of the Bayes filter
- Using the entire state history, we eliminate the need to integrate during the prediction
- However, we still need to compute a normalization constant
- We also haven't said how we'll extract values
- We handle both of these issues using *maximum likelihood estimation*