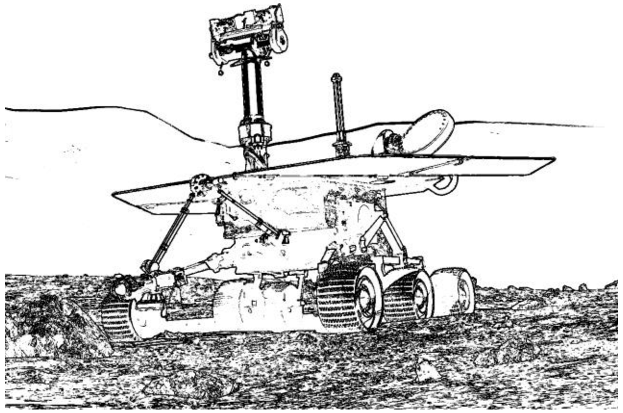


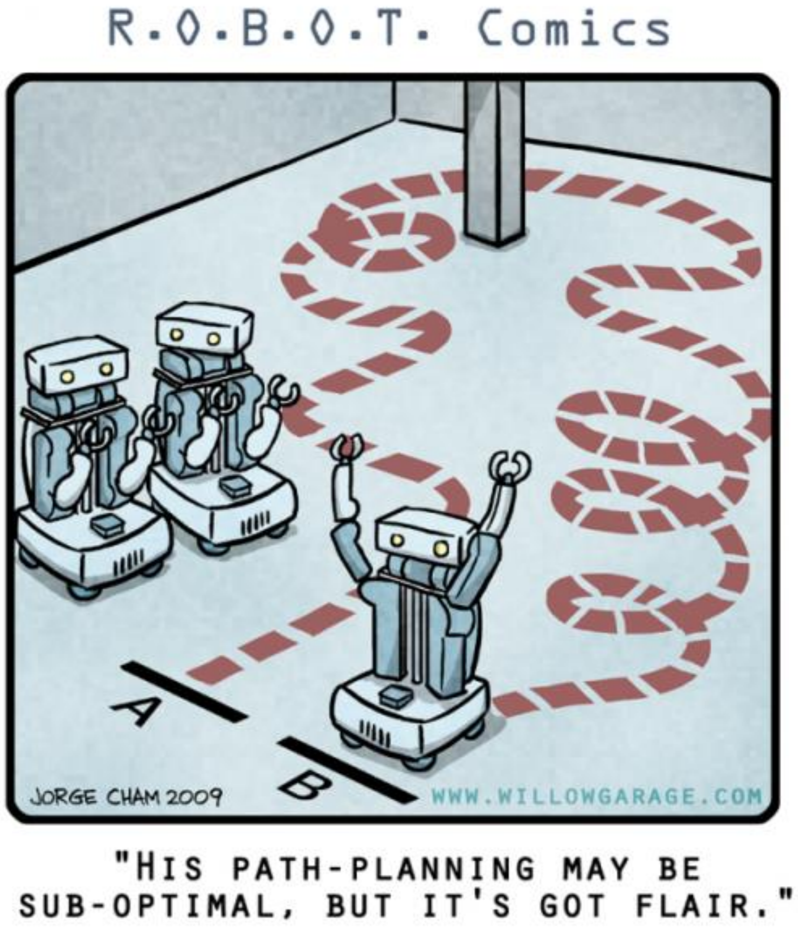
CSC477
Introduction to Mobile Robotics

Florian Shkurti

Week #7: Least Squares Estimation and GraphSLAM

Today's agenda

- Least Squares Estimation
- Maximum Likelihood Estimation (MLE)
- Maximum a Posteriori Estimation (MAP)
- Bayesian Estimation
- GraphSLAM



Estimating parameters of probability models

- In the occupancy grid mapping problem we wanted to compute $p(\mathbf{m}|\mathbf{z}_{1:t}, \mathbf{x}_{1:t})$ over all possible maps.

- We can see this problem as a specific instance within a category of problems where we are given data (observations) and we want to "explain" or fit the data using a parametric function.

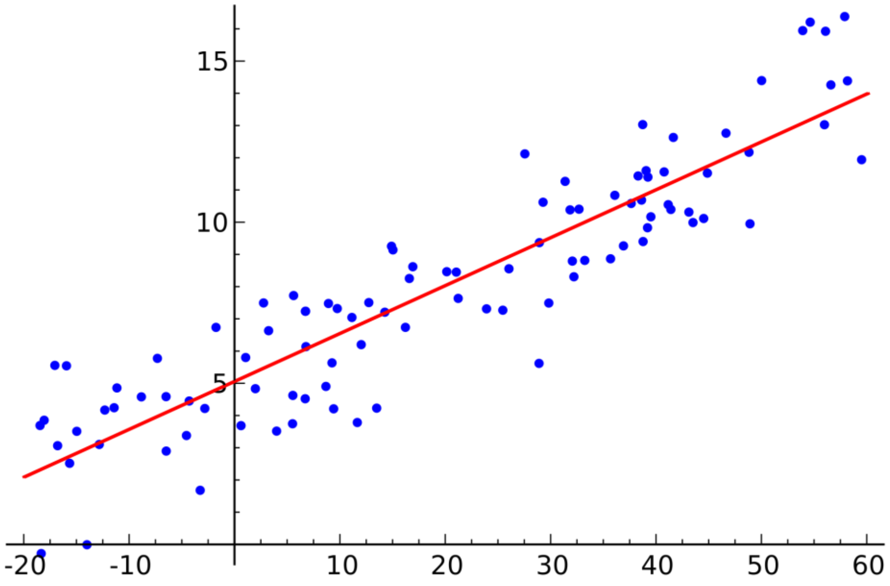
Estimating parameters of probability models

- In the occupancy grid mapping problem we wanted to compute $p(\mathbf{m}|\mathbf{z}_{1:t}, \mathbf{x}_{1:t})$ over all possible maps.

- We can see this problem as a specific instance within a category of problems where we are given data (observations) and we want to “explain” or fit the data using a parametric function.

- There are typically three ways to work with this type of problems:
  1. Maximum Likelihood parameter estimation (MLE)
     - Least Squares
  2. Maximum A Posteriori (MAP) parameter estimation
  3. Bayesian parameter distribution estimation

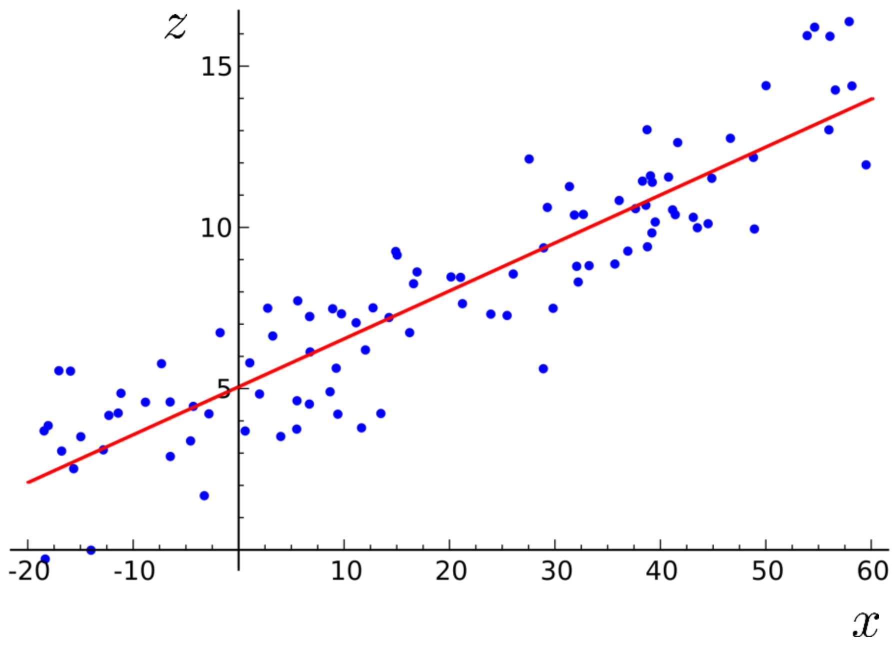
Least Squares Parameter Estimation



We are given data points $(\mathbf{x}_1, \mathbf{z}_1), ..., (\mathbf{x}_N, \mathbf{z}_N)$

We **think** that the data was generated by a parametric function $\mathbf{z} = \mathbf{h}(\boldsymbol{\theta}, \mathbf{x})$

Least Squares Parameter Estimation

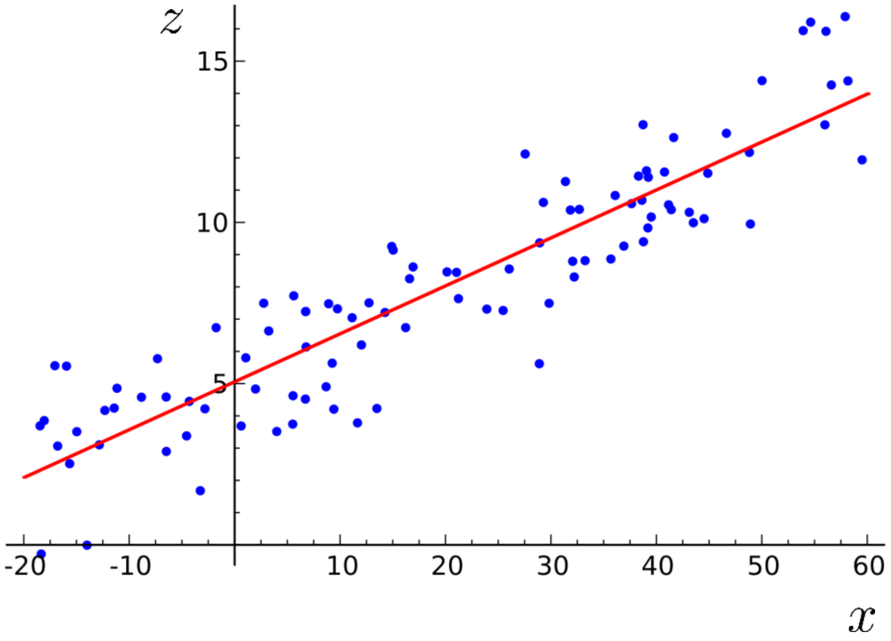


We are given data points $(\mathbf{x}_1, \mathbf{z}_1), ..., (\mathbf{x}_N, \mathbf{z}_N)$

We **think** that the data was generated by a parametric function $\mathbf{z} = \mathbf{h}(\boldsymbol{\theta}, \mathbf{x})$

Example: we think that the 2D data was generated by a line $z = \theta_0 + \theta_1 x$ whose parameters we do not know, and was corrupted by noise.

Least Squares Parameter Estimation



Example: we think that the 2D data was generated by a line $z = \theta_0 + \theta_1 x$ whose parameters we do not know.

We are given data points $(\mathbf{x}_1, \mathbf{z}_1), ..., (\mathbf{x}_N, \mathbf{z}_N)$

We **think** that the data was generated by a parametric function $\mathbf{z} = \mathbf{h}(\boldsymbol{\theta}, \mathbf{x})$

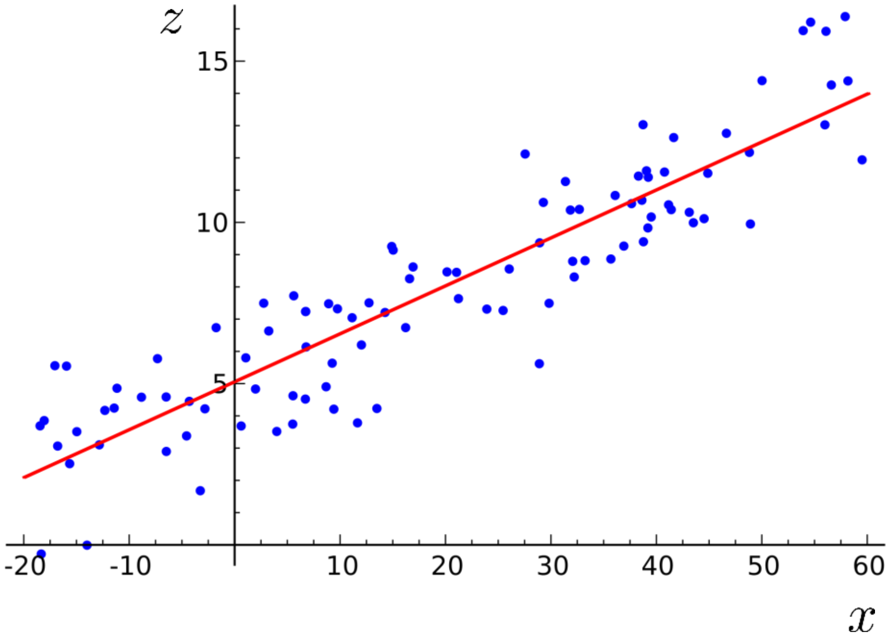
This parametric model will have a fitting error:

$$e(\boldsymbol{\theta}) = \sum_{i=1}^{N} ||\mathbf{z}_i - \mathbf{h}(\boldsymbol{\theta}, \mathbf{x}_i)||^2$$

The least-squares estimator is:

$$\boldsymbol{\theta}_{LS} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\ e(\boldsymbol{\theta})$$

Linear Least Squares Parameter Estimation



Example: we think that the 2D data was generated by a line $z = \theta_0 + \theta_1 x$ whose parameters we do not know.

We are given data points $(\mathbf{x}_1, \mathbf{z}_1), ..., (\mathbf{x}_N, \mathbf{z}_N)$

We **think** that the data was generated by a linear parametric function $\mathbf{z} = \mathbf{h}(\boldsymbol{\theta}, \mathbf{x}) = \mathbf{H}_{\mathbf{x}}\boldsymbol{\theta}$
where $\mathbf{H}_{\mathbf{x}}$ is a matrix whose elements depend on $\mathbf{x}$

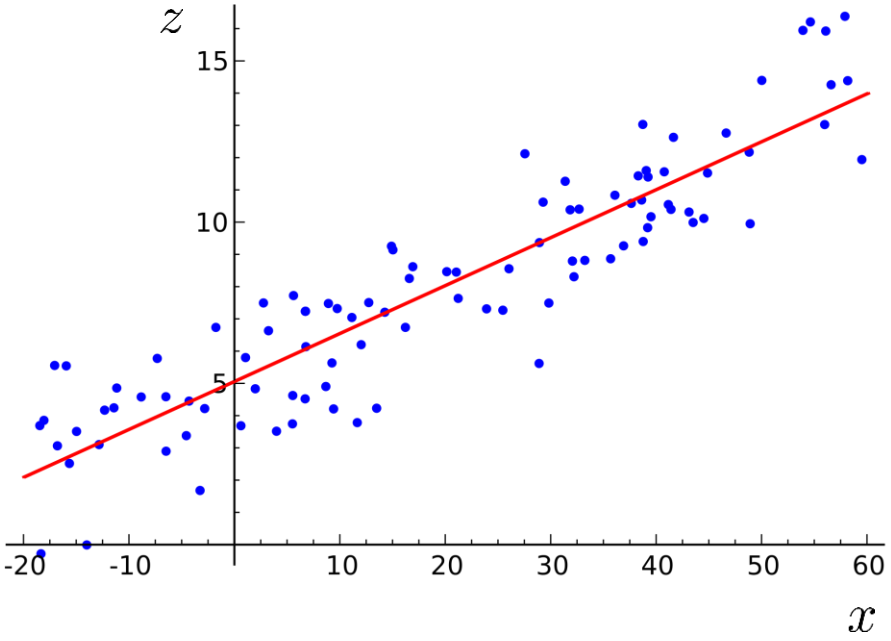
This parametric model will have a fitting error:

$$e(\boldsymbol{\theta}) = \sum_{i=1}^{N} ||\mathbf{z}_i - \mathbf{H}_{\mathbf{x}_i}\boldsymbol{\theta}||^2$$

The least-squares estimator is:

$$\boldsymbol{\theta}_{LS} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, e(\boldsymbol{\theta})$$

Linear Least Squares Parameter Estimation



Example: we think that the 2D data was generated by a line $z = \theta_0 + \theta_1 x$ whose parameters we do not know.

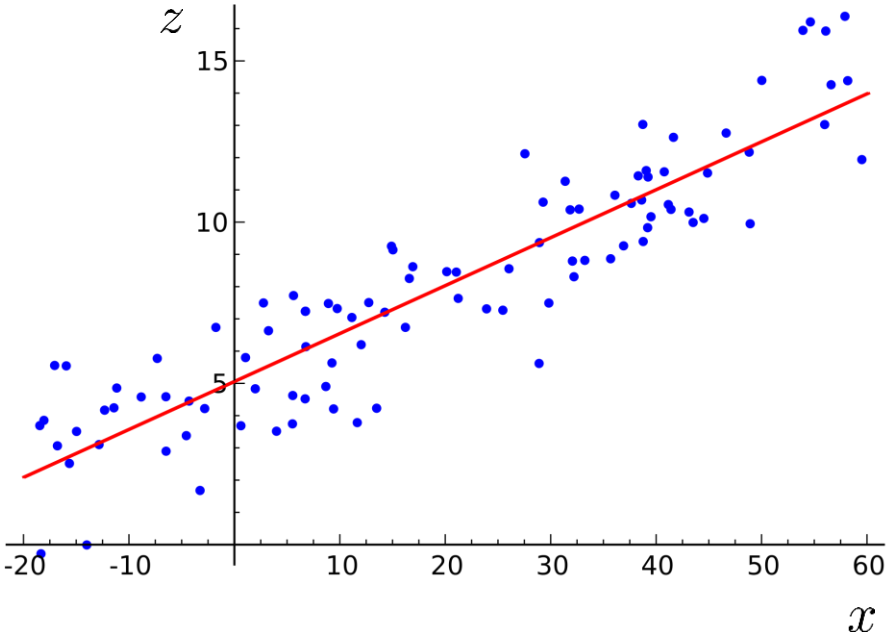
We are given data points $(\mathbf{x}_1, \mathbf{z}_1), ..., (\mathbf{x}_N, \mathbf{z}_N)$

We **think** that the data was generated by a linear parametric function $\mathbf{z} = \mathbf{h}(\boldsymbol{\theta}, \mathbf{x}) = \mathbf{H}_{\mathbf{x}}\boldsymbol{\theta}$

This parametric model will have a fitting error:

$$\begin{aligned} e(\boldsymbol{\theta}) &= \sum_{i=1}^{N} ||\mathbf{z}_i - \mathbf{H}_{\mathbf{x}_i}\boldsymbol{\theta}||^2 \\ &= \sum_{i=1}^{N} \mathbf{z}_i^T \mathbf{z}_i - 2\boldsymbol{\theta}^T \mathbf{H}_{\mathbf{x}_i}^T \mathbf{z}_i + \boldsymbol{\theta}^T \mathbf{H}_{\mathbf{x}_i}^T \mathbf{H}_{\mathbf{x}_i} \boldsymbol{\theta} \end{aligned}$$

Linear Least Squares Parameter Estimation



Example: we think that the 2D data was generated by a line $z = \theta_0 + \theta_1 x$ whose parameters we do not know.

We are given data points $(\mathbf{x}_1, \mathbf{z}_1), ..., (\mathbf{x}_N, \mathbf{z}_N)$

We **think** that the data was generated by a linear parametric function $\mathbf{z} = \mathbf{h}(\boldsymbol{\theta}, \mathbf{x}) = \mathbf{H}_{\mathbf{x}}\boldsymbol{\theta}$

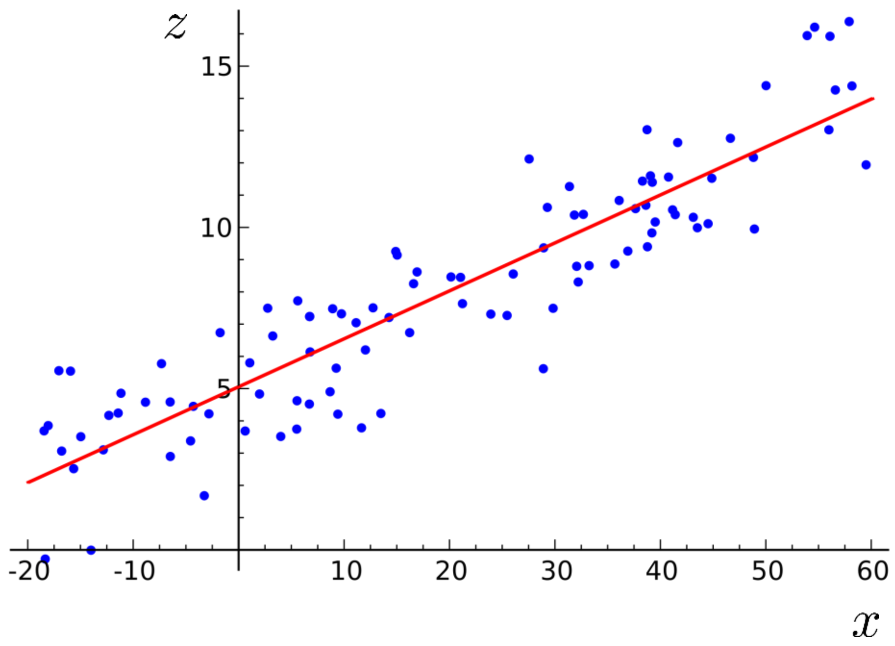
This parametric model will have a fitting error:

$$
\begin{aligned}
e(\boldsymbol{\theta}) &= \sum_{i=1}^{N} ||\mathbf{z}_i - \mathbf{H}_{\mathbf{x}_i}\boldsymbol{\theta}||^2 \\
&= \sum_{i=1}^{N} \mathbf{z}_i^T\mathbf{z}_i - 2\boldsymbol{\theta}^T\mathbf{H}_{\mathbf{x}_i}^T\mathbf{z}_i + \boldsymbol{\theta}^T\mathbf{H}_{\mathbf{x}_i}^T\mathbf{H}_{\mathbf{x}_i}\boldsymbol{\theta}
\end{aligned}
$$

The least-squares estimator minimizes the error:

$$\frac{\partial e(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \mathbf{0} \Leftrightarrow -2\sum_{i=1}^{N} \mathbf{H}_{\mathbf{x}_i}^T\mathbf{z}_i + 2\mathbf{H}_{\mathbf{x}_i}^T\mathbf{H}_{\mathbf{x}_i}\boldsymbol{\theta} = \mathbf{0} \Leftrightarrow \left[\sum_{i=1}^{N} \mathbf{H}_{\mathbf{x}_i}^T\mathbf{H}_{\mathbf{x}_i}\right]\boldsymbol{\theta} = \sum_{i=1}^{N} \mathbf{H}_{\mathbf{x}_i}^T\mathbf{z}_i$$

Linear Least Squares Parameter Estimation



Example: we think that the 2D data was generated by a line $z = \theta_0 + \theta_1 x$ whose parameters we do not know.

We are given data points $(\mathbf{x}_1, \mathbf{z}_1), ..., (\mathbf{x}_N, \mathbf{z}_N)$

We **think** that the data was generated by a linear parametric function $\mathbf{z} = \mathbf{h}(\boldsymbol{\theta}, \mathbf{x}) = \mathbf{H}_{\mathbf{x}}\boldsymbol{\theta}$

This parametric model will have a fitting error:

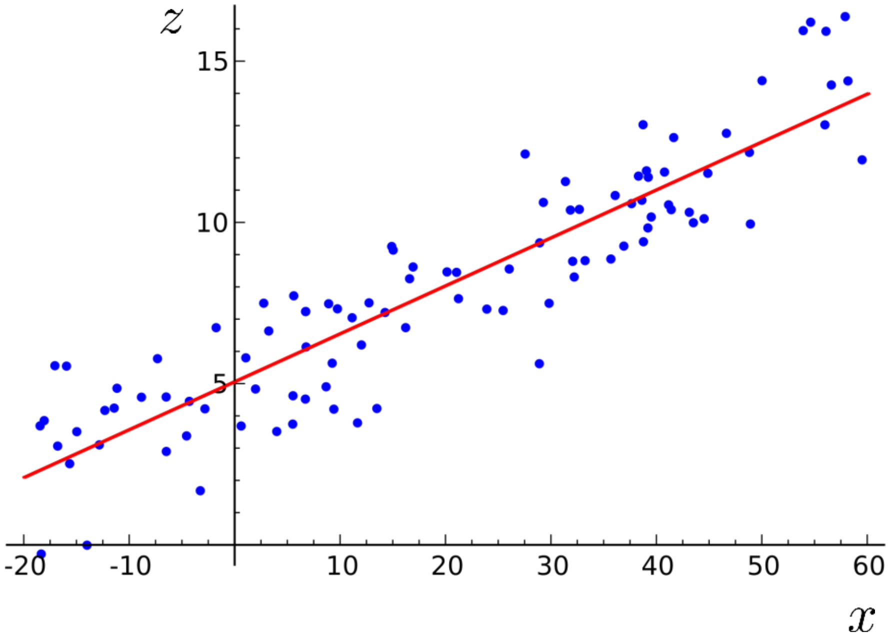
$$\begin{aligned} e(\boldsymbol{\theta}) &= \sum_{i=1}^{N} ||\mathbf{z}_i - \mathbf{H}_{\mathbf{x}_i}\boldsymbol{\theta}||^2 \\ &= \sum_{i=1}^{N} \mathbf{z}_i^T\mathbf{z}_i - 2\boldsymbol{\theta}^T\mathbf{H}_{\mathbf{x}_i}^T\mathbf{z}_i + \boldsymbol{\theta}^T\mathbf{H}_{\mathbf{x}_i}^T\mathbf{H}_{\mathbf{x}_i}\boldsymbol{\theta} \end{aligned}$$

The least-squares estimator minimizes the error:

$$\frac{\partial e(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \mathbf{0} \Leftrightarrow -2\sum_{i=1}^{N} \mathbf{H}_{\mathbf{x}_i}^T\mathbf{z}_i + 2\mathbf{H}_{\mathbf{x}_i}^T\mathbf{H}_{\mathbf{x}_i}\boldsymbol{\theta} = \mathbf{0} \Leftrightarrow \left[\sum_{i=1}^{N} \mathbf{H}_{\mathbf{x}_i}^T\mathbf{H}_{\mathbf{x}_i}\right]\boldsymbol{\theta} = \sum_{i=1}^{N} \mathbf{H}_{\mathbf{x}_i}^T\mathbf{z}_i$$

$$\boldsymbol{\theta}_{LS} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, e(\boldsymbol{\theta}) \Leftrightarrow \left[\sum_{i=1}^{N} \mathbf{H}_{\mathbf{x}_i}^T\mathbf{H}_{\mathbf{x}_i}\right]\boldsymbol{\theta}_{LS} = \sum_{i=1}^{N} \mathbf{H}_{\mathbf{x}_i}^T\mathbf{z}_i$$

Example #1: Linear Least Squares



Example: we think that the 2D data was generated by a line $z = \theta_0 + \theta_1 x$ whose parameters we do not know.

We are given 2D data points $(x_1, z_1), ..., (x_N, z_N)$

We **think** that the data was generated by a linear parametric function $z = h(\boldsymbol{\theta}, x) = [1 \quad x]\boldsymbol{\theta} = \theta_0 + \theta_1 x$

This parametric model will have a fitting error:

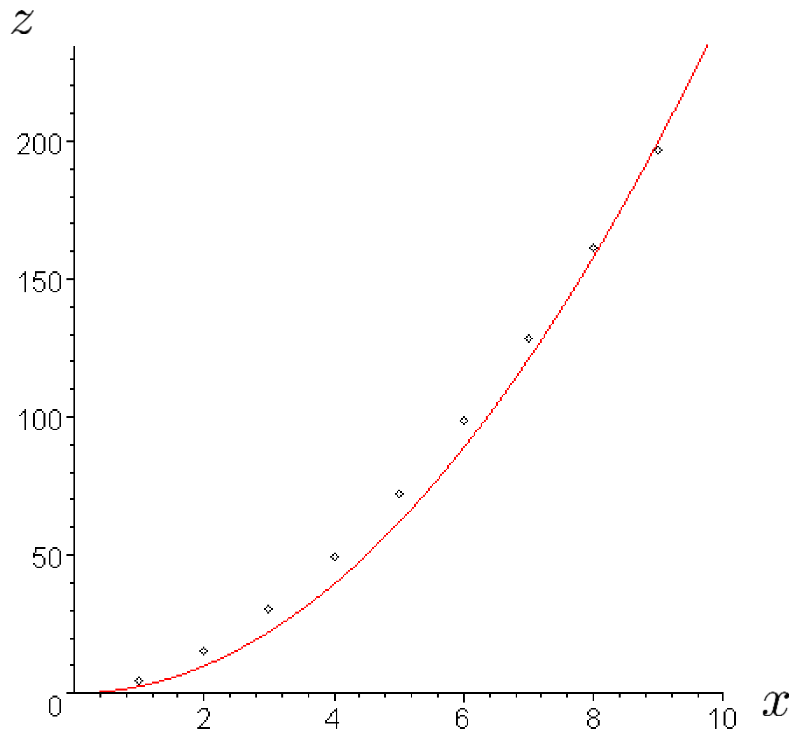
$$e(\theta_0, \theta_1) \quad = \quad \sum_{i=1}^{N} (z_i - \theta_0 - \theta_1 x_i)^2$$

The least-squares estimator minimizes the error:

$$\boldsymbol{\theta}_{LS} = \underset{\theta_0, \theta_1}{\text{argmin}}\ e(\theta_0, \theta_1) \Leftrightarrow \left[\sum_{i=1}^{N} \begin{bmatrix} 1 \\ x_i \end{bmatrix} [1 \quad x_i] \right] \boldsymbol{\theta}_{LS} = \sum_{i=1}^{N} \begin{bmatrix} 1 \\ x_i \end{bmatrix} z_i$$

Which is a linear system of 2 equations. If we have at least two data points we can solve for $\boldsymbol{\theta}_{LS}$ to define the line.

Example #2: Linear Least Squares



Example: we think that the 2D data was generated by a quadratic $z = \theta_0 + \theta_1 x + \theta_2 x^2$ whose parameters we do not know.

We are given 2D data points $(x_1, z_1), ..., (x_N, z_N)$

We **think** that the data was generated by a linear parametric function $z = h(\boldsymbol{\theta}, x) = [1 \quad x \quad x^2]\boldsymbol{\theta} = \theta_0 + \theta_1 x + \theta_2 x^2$

This parametric model will have a fitting error:

$$e(\theta_0, \theta_1, \theta_2) \quad = \quad \sum_{i=1}^{N} (z_i - \theta_0 - \theta_1 x_i - \theta_2 x_i^2)^2$$

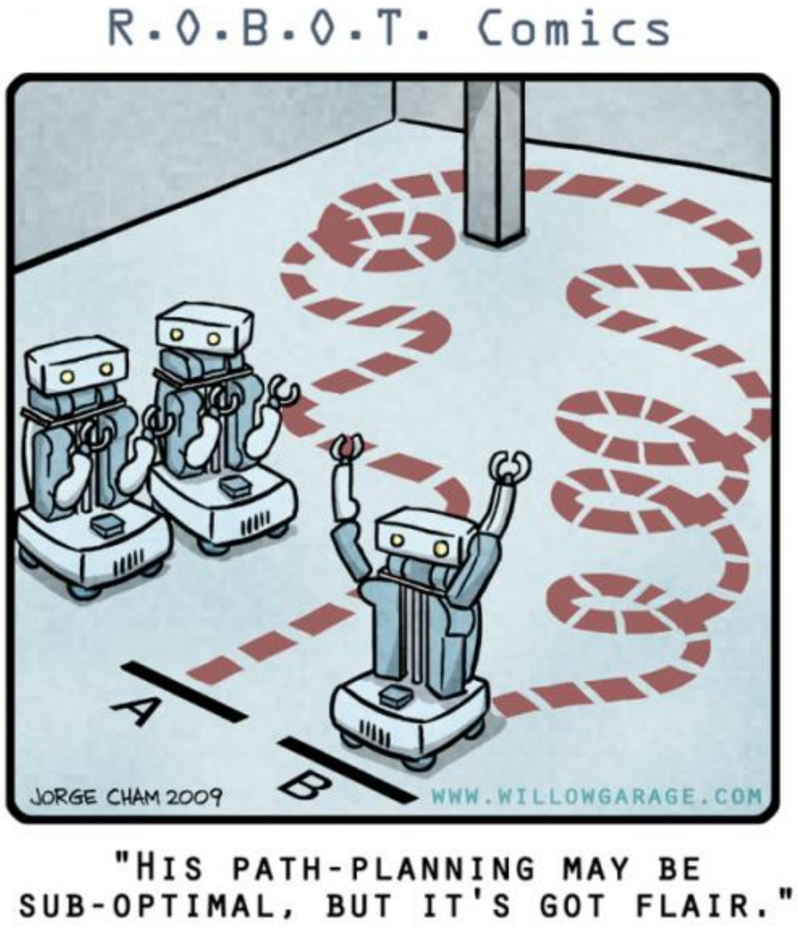
The least-squares estimator minimizes the error:

$$\boldsymbol{\theta}_{LS} = \underset{\theta_0, \theta_1, \theta_2}{\operatorname{argmin}}\ e(\theta_0, \theta_1, \theta_2) \Leftrightarrow \left[\sum_{i=1}^{N} \begin{bmatrix} 1 \\ x_i \\ x_i^2 \end{bmatrix} [1 \quad x_i \quad x_i^2] \right] \boldsymbol{\theta}_{LS} = \sum_{i=1}^{N} \begin{bmatrix} 1 \\ x_i \\ x_i^2 \end{bmatrix} z_i$$

Which is a linear system of 3 equations. If we have at least three data points we can solve for $\boldsymbol{\theta}_{LS}$ to define the quadratic.

Today's agenda

- Least Squares Estimation
- Maximum Likelihood Estimation (MLE)
- Maximum a Posteriori Estimation (MAP)
- Bayesian Estimation
- GraphSLAM



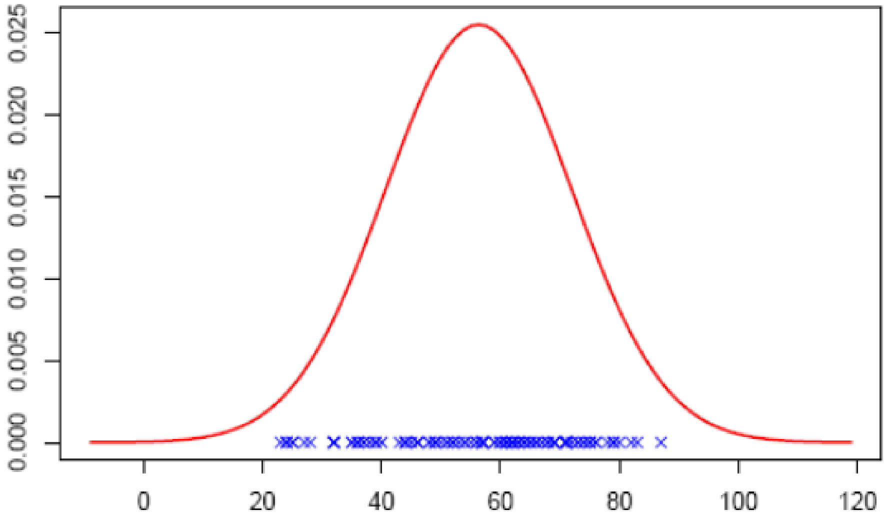
Estimating parameters of probability models

- In the occupancy grid mapping problem we wanted to compute $p(\mathbf{m}|\mathbf{z}_{1:t}, \mathbf{x}_{1:t})$ over all possible maps.

- We can see this problem as a specific instance within a category of problems where we are given data (observations) and we want to "explain" or fit the data using a parametric function.

- There are typically three ways to work with this type of problems:
    1. Maximum Likelihood parameter estimation (MLE)
        - Least Squares
    2. Maximum A Posteriori (MAP) parameter estimation
    3. Bayesian parameter distribution estimation

Maximum Likelihood Parameter Estimation



We are given data points $\mathbf{d}_{1:N} = \mathbf{d}_1, ..., \mathbf{d}_N$

We **think** the data has been generated from a probability distribution $p(\mathbf{d}_{1:N}|\boldsymbol{\theta})$

We want to find the parameter of the model that maximizes the likelihood function of the data

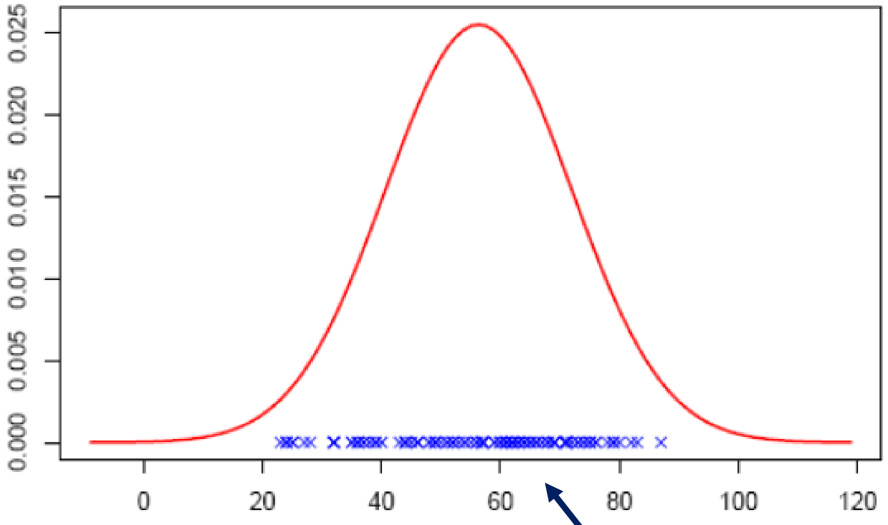
$$L(\boldsymbol{\theta}) = p(\mathbf{d}_{1:N}|\boldsymbol{\theta})$$

which is a function of theta, **not** a probability distribution.

$$\boldsymbol{\theta}_{MLE} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, p(\mathbf{d}_{1:N}|\boldsymbol{\theta})$$

Maximum Likelihood Parameter Estimation

Data points



$$\boldsymbol{\theta}_{MLE} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, p(\mathbf{d}_{1:N}|\boldsymbol{\theta})$$

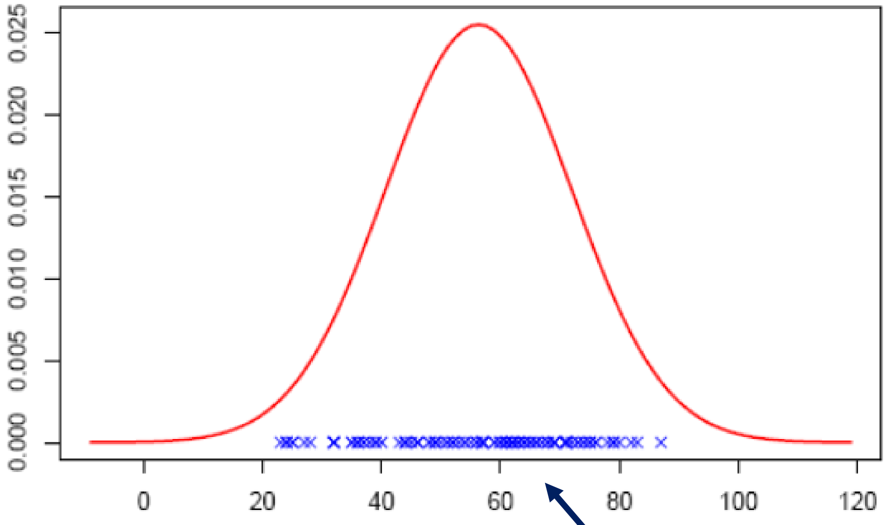
Find the parameters of the model that maximize the likelihood function of the data

$$L(\boldsymbol{\theta}) = p(\mathbf{d}_{1:N}|\boldsymbol{\theta})$$

which is a function of theta, **not** a probability distribution.

Example: assume we know that 1D data points were generated independently from a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, but we don't know the mean and variance. The likelihood function of the data is

Maximum Likelihood Parameter Estimation



Data points

$$\boldsymbol{\theta}_{MLE} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, p(\mathbf{d}_{1:N}|\boldsymbol{\theta})$$

Find the parameters of the model that maximize the likelihood function of the data

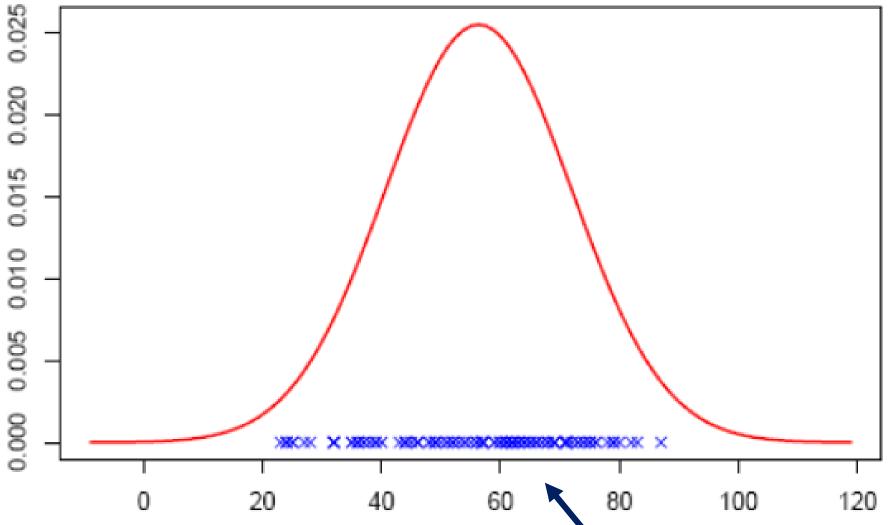
$$L(\boldsymbol{\theta}) = p(\mathbf{d}_{1:N}|\boldsymbol{\theta})$$

which is a function of theta, **not** a probability distribution.

Example: assume we know that 1D data points were generated independently from a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, but we don't know the mean and variance. The likelihood function of the data is

$$L(\mu, \sigma) = p(\mathbf{d}_{1:N}|\mu, \sigma) = \prod_{i=1}^{N} p(d_i|\mu, \sigma) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp(-0.5(d_i - \mu)^2/\sigma^2)$$

Maximum Likelihood Parameter Estimation



Data points

$$\boldsymbol{\theta}_{MLE} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\ p(\mathbf{d}_{1:N}|\boldsymbol{\theta})$$

Find the parameters of the model that maximize the likelihood function of the data

$$L(\boldsymbol{\theta}) = p(\mathbf{d}_{1:N}|\boldsymbol{\theta})$$

which is a function of theta, **not** a probability distribution.

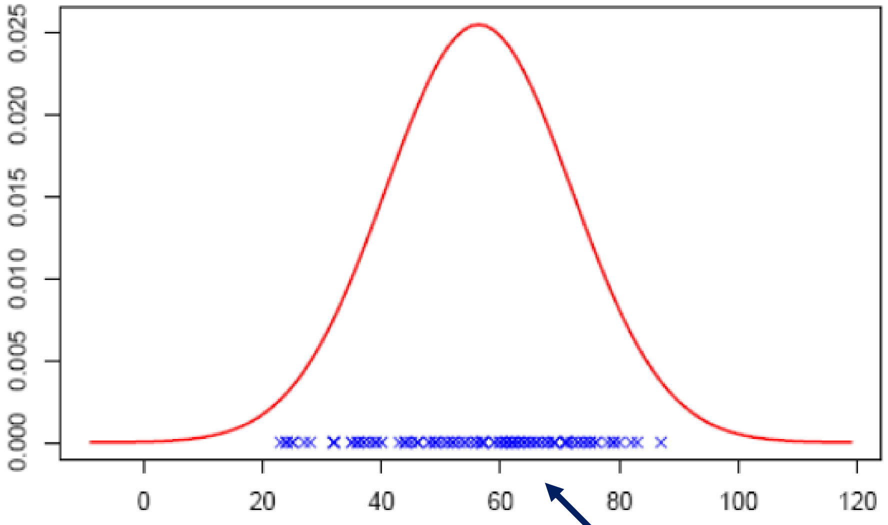
Example: assume we know that 1D data points were generated independently from a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, but we don't know the mean and variance. The likelihood function of the data is

$$L(\mu,\sigma) = p(\mathbf{d}_{1:N}|\mu,\sigma) = \prod_{i=1}^{N} p(d_i|\mu,\sigma) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp(-0.5(d_i-\mu)^2/\sigma^2)$$

And the maximum-likelihood parameter estimates are

$$(\mu,\sigma)_{MLE} = \underset{\mu,\sigma}{\operatorname{argmax}}\ p(\mathbf{d}_{1:N}|\mu,\sigma) = \underset{\mu,\sigma}{\operatorname{argmax}}\ \log p(\mathbf{d}_{1:N}|\mu,\sigma) = \underset{\mu,\sigma}{\operatorname{argmax}} \sum_{i=1}^{N} \log p(d_i|\mu,\sigma)$$

Maximum Likelihood Parameter Estimation



Data points

$$\boldsymbol{\theta}_{MLE} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\; p(\mathbf{d}_{1:N}|\boldsymbol{\theta})$$

Find the parameters of the model that maximize the likelihood function of the data

$$L(\boldsymbol{\theta}) = p(\mathbf{d}_{1:N}|\boldsymbol{\theta})$$

which is a function of theta, **not** a probability distribution.

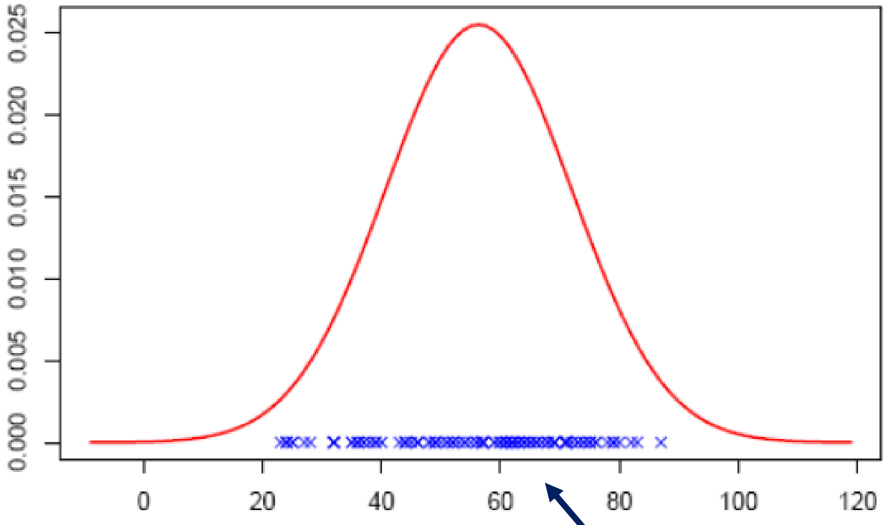
Example: assume we know that 1D data points were generated independently from a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, but we don't know the mean and variance. The likelihood function of the data is

$$L(\mu,\sigma) = p(\mathbf{d}_{1:N}|\mu,\sigma) = \prod_{i=1}^{N} p(d_i|\mu,\sigma) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma}\exp(-0.5(d_i-\mu)^2/\sigma^2)$$

And the maximum-likelihood parameter estimates are

$$(\mu,\sigma)_{MLE} = \underset{\mu,\sigma}{\operatorname{argmax}} \sum_{i=1}^{N} \log p(d_i|\mu,\sigma) = \underset{\mu,\sigma}{\operatorname{argmax}} \left[-N\log(\sqrt{2\pi}\sigma) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(d_i-\mu)^2 \right]$$

Maximum Likelihood Parameter Estimation



Data points

$$\boldsymbol{\theta}_{MLE} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, p(\mathbf{d}_{1:N}|\boldsymbol{\theta})$$

Find the parameters of the model that maximize the likelihood function of the data

$$L(\boldsymbol{\theta}) = p(\mathbf{d}_{1:N}|\boldsymbol{\theta})$$

which is a function of theta, **not** a probability distribution.

Example: assume we know that 1D data points were generated independently from a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, but we don't know the mean and variance. The likelihood function of the data is

$$L(\mu,\sigma) = p(\mathbf{d}_{1:N}|\mu,\sigma) = \prod_{i=1}^{N} p(d_i|\mu,\sigma) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma}\exp(-0.5(d_i-\mu)^2/\sigma^2)$$

And the maximum-likelihood parameter estimates are

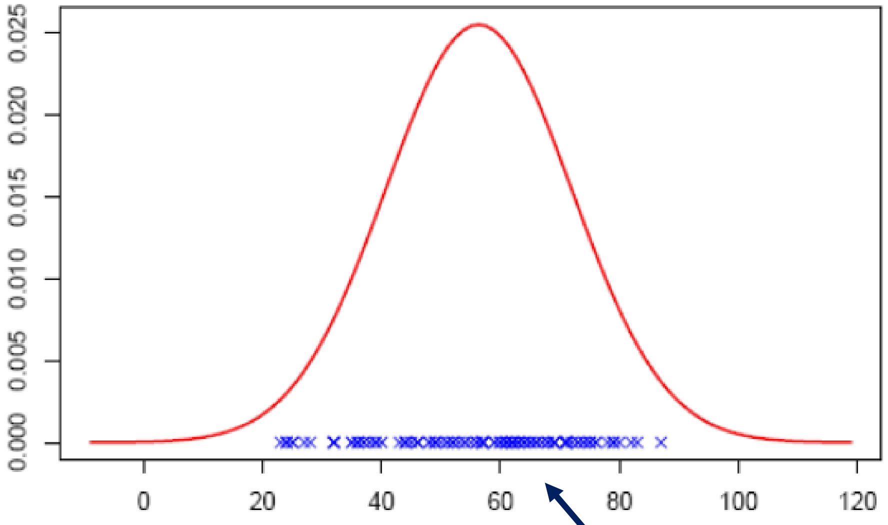
$$(\mu,\sigma)_{MLE} = \underset{\mu,\sigma}{\operatorname{argmax}} \sum_{i=1}^{N} \log p(d_i|\mu,\sigma) = \underset{\mu,\sigma}{\operatorname{argmax}} \left[-N\log(\sqrt{2\pi}\sigma) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(d_i-\mu)^2 \right]$$

Set partial derivatives w.r.t. μ and σ to zero

$$\mu_{MLE} = \sum_{i=1}^{N} d_i/N$$

$$\sigma^2_{MLE} = \frac{1}{N}\sum_{i=1}^{N}(d_i-\mu_{MLE})^2$$

Least Squares as Maximum Likelihood



$$\boldsymbol{\theta}_{MLE} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, p(\mathbf{d}_{1:N}|\boldsymbol{\theta})$$

Find the parameters of the model that maximize the likelihood function of the data

$$L(\boldsymbol{\theta}) = p(\mathbf{d}_{1:N}|\boldsymbol{\theta})$$

which is a function of theta, **not** a probability distribution.

Example: assume we know that 1D data points were generated independently from a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, but we don't know the mean and variance. The likelihood function of the data is

$$L(\mu,\sigma) = p(\mathbf{d}_{1:N}|\mu,\sigma) = \prod_{i=1}^{N} p(d_i|\mu,\sigma) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma}\exp(-0.5(d_i-\mu)^2/\sigma^2)$$

And the maximum-likelihood parameter estimates are

$$(\mu,\sigma)_{MLE} = \underset{\mu,\sigma}{\operatorname{argmax}} \sum_{i=1}^{N} \log p(d_i|\mu,\sigma) = \underset{\mu,\sigma}{\operatorname{argmax}} \left[-N\log(\sqrt{2\pi}\sigma) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(d_i-\mu)^2 \right]$$

Least squares estimation occurs from maximum likelihood with Gaussian models of data

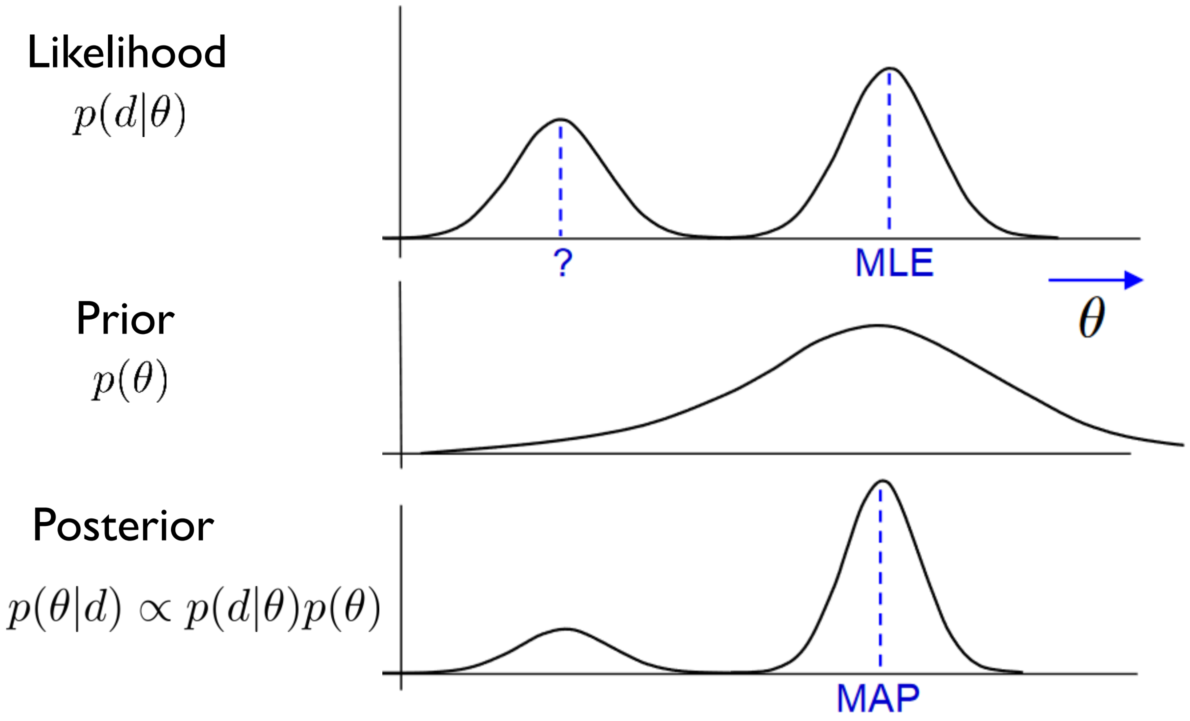
Estimating parameters of probability models

- In the occupancy grid mapping problem we wanted to compute $p(\mathbf{m}|\mathbf{z}_{1:t}, \mathbf{x}_{1:t})$ over all possible maps.

- We can see this problem as a specific instance within a category of problems where we are given data (observations) and we want to "explain" or fit the data using a parametric function.

- There are typically three ways to work with this type of problems:
  1. Maximum Likelihood parameter estimation (MLE)
     - Least Squares
  2. Maximum A Posteriori (MAP) parameter estimation
  3. Bayesian parameter distribution estimation

Maximum A Posteriori Parameter Estimation



$$
\begin{aligned}
\boldsymbol{\theta}_{MAP} &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, p(\boldsymbol{\theta}|\mathbf{d}_{1:N}) \\
&= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left[\frac{p(\mathbf{d}_{1:N}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{d}_{1:N})}\right] \\
&= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left[p(\mathbf{d}_{1:N}|\boldsymbol{\theta})p(\boldsymbol{\theta})\right]
\end{aligned}
$$

Almost the same as MLE, but with a prior distribution on the parameters

Estimating parameters of probability models

- In the occupancy grid mapping problem we wanted to compute $p(\mathbf{m}|\mathbf{z}_{1:t}, \mathbf{x}_{1:t})$ over all possible maps.

- We can see this problem as a specific instance within a category of problems where we are given data (observations) and we want to "explain" or fit the data using a parametric function.

- There are typically three ways to work with this type of problems:
  1. Maximum Likelihood parameter estimation (MLE)
     - Least Squares
  2. Maximum A Posteriori (MAP) parameter estimation
  3. Bayesian parameter distribution estimation

Bayesian parameter estimation

- Both MLE and MAP estimators give you a single **point estimate**.
- But there might be many parameters that are compatible with the data.
- Instead of point estimates, compute a **distribution of estimates** that explain the data

- Bayesian parameter estimation:

$$p(\boldsymbol{\theta}|\mathbf{d}_{1:N}) = \frac{p(\mathbf{d}_{1:N}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{d}_{1:N})}$$

The probability of the data is usually hard to compute. But it does not depend on the parameter theta, so it is treated as a normalizing factor, and we can still compute how the posterior varies with theta.

Bayesian parameter estimation

- Both MLE and MAP estimators give you a single **point estimate**.
- But there might be many parameters that are compatible with the data.
- Instead of point estimates, compute a **distribution of estimates** that explain the data

- Bayesian parameter estimation:

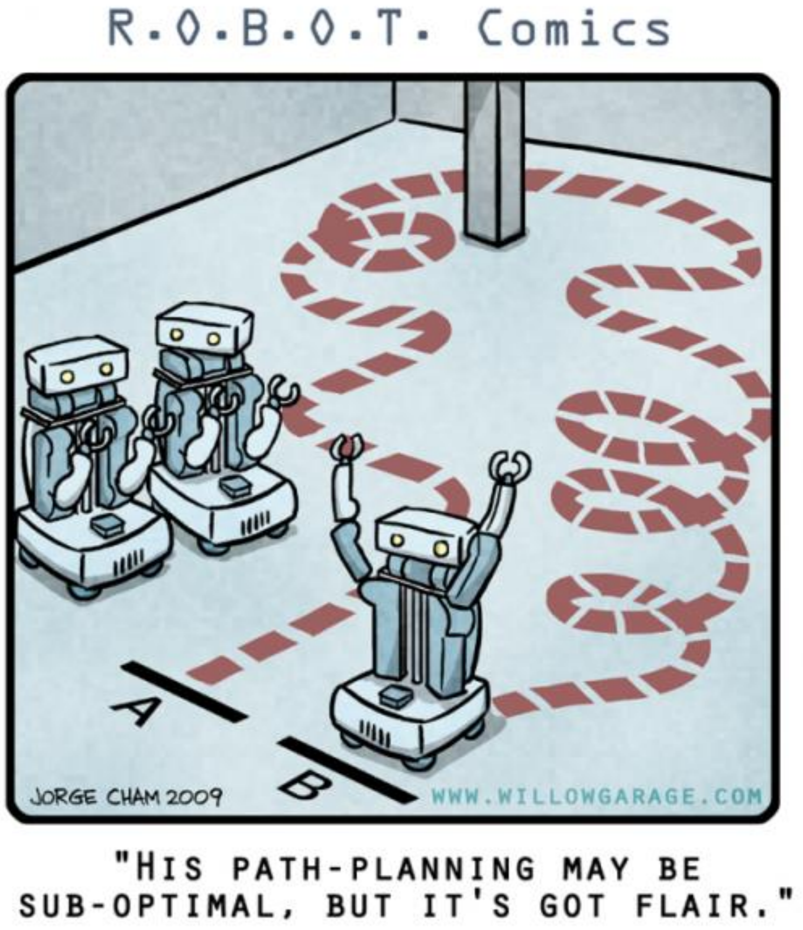
$$p(\boldsymbol{\theta}|\mathbf{d}_{1:N}) = \frac{p(\mathbf{d}_{1:N}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{d}_{1:N})}$$

- This is what we used in occupancy grid mapping, when we approximated

$$p(\mathbf{m}|\mathbf{z}_{1:t}, \mathbf{x}_{1:t})$$

Today's agenda

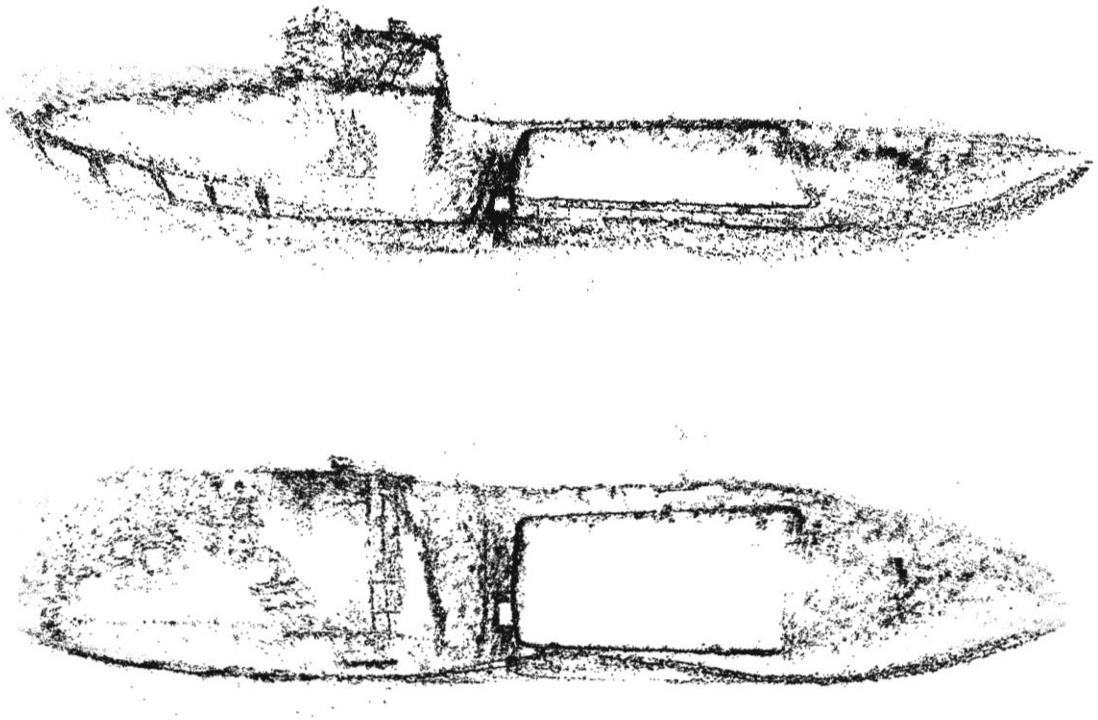
- Least Squares Estimation
- Maximum Likelihood Estimation (MLE)
- Maximum a Posteriori Estimation (MAP)
- Bayesian Estimation
- GraphSLAM

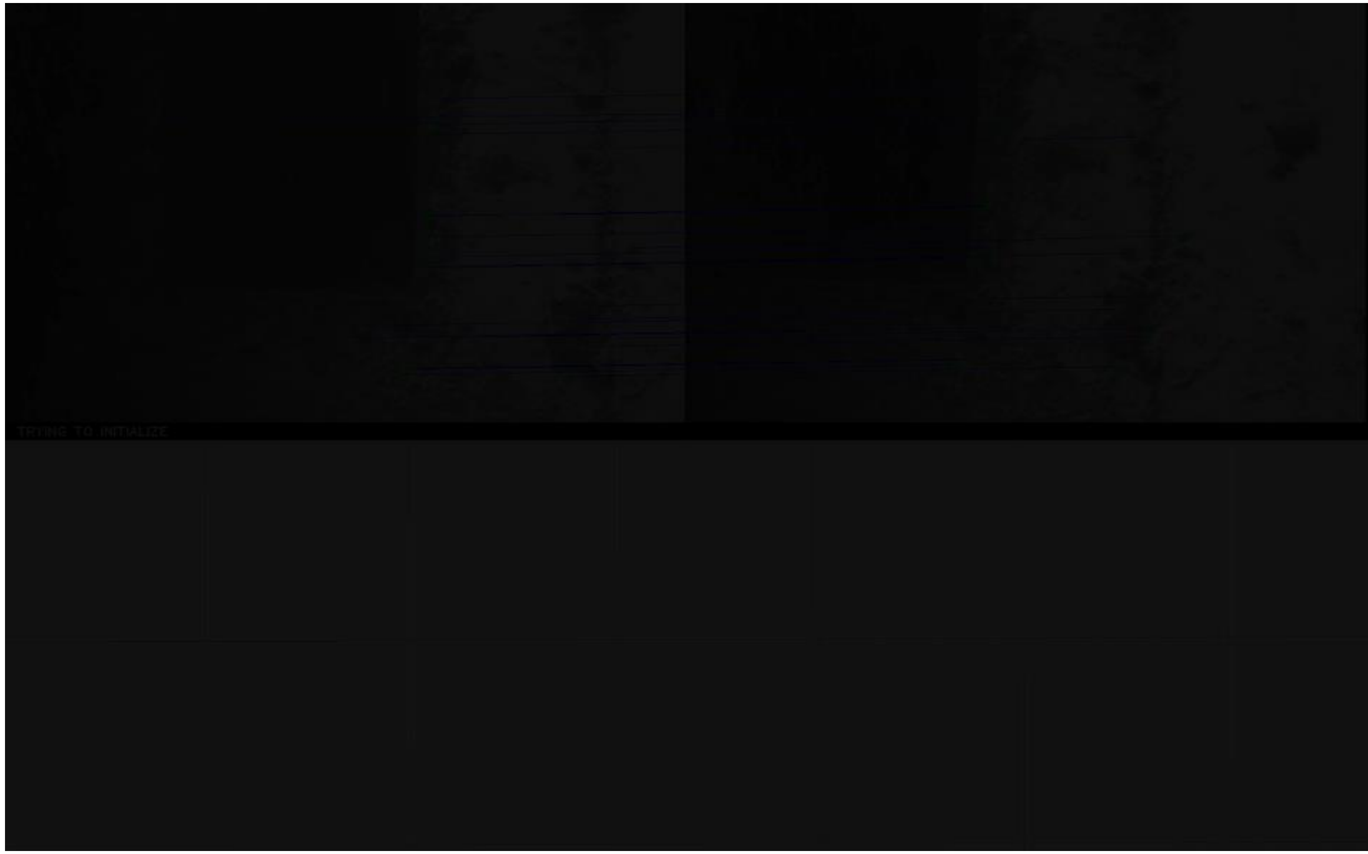


Goal

- Enable a robot to simultaneously build a map of its environment and estimate where it is in that map.
- This is called SLAM (Simultaneous Localization And Mapping)
- Today we are going to look at the batch version, i.e. collect all measurements and controls, and later form an estimate of the states and the map.
- We are going to solve SLAM using least squares

Examples of SLAM systems

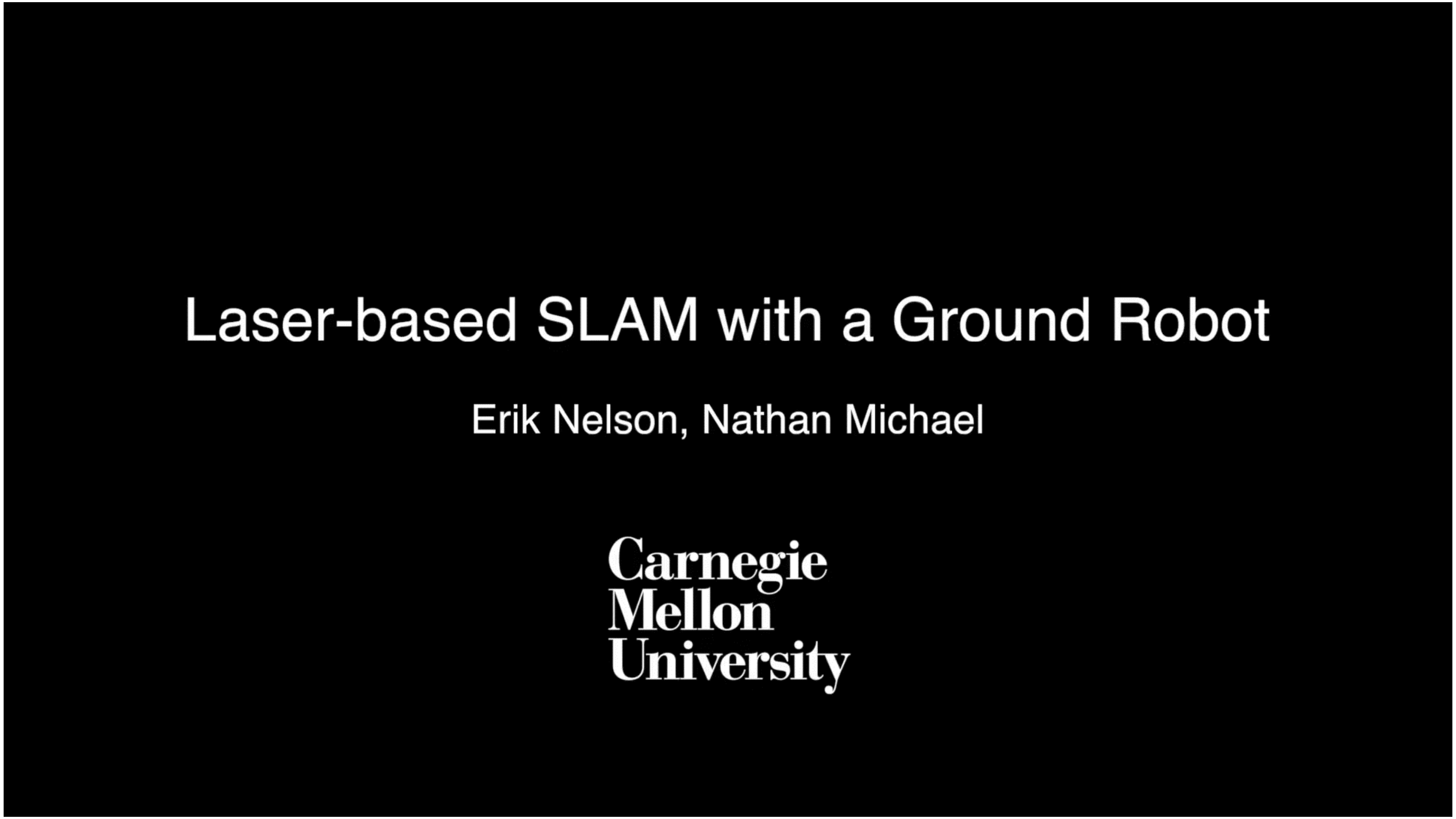




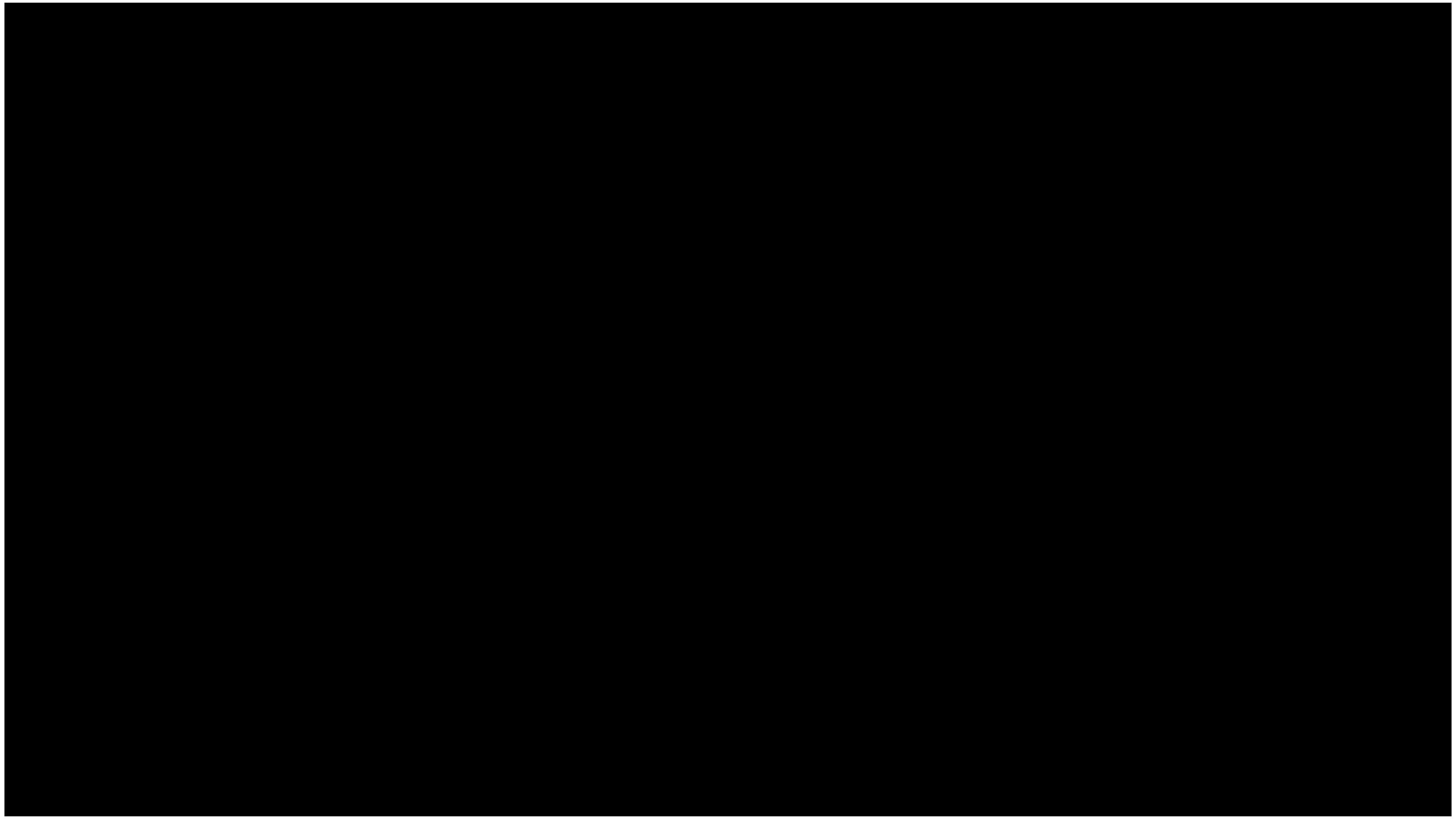
MORESLAM system, McGill, 2016

MORESLAM system, McGill, 2016

Examples of SLAM systems



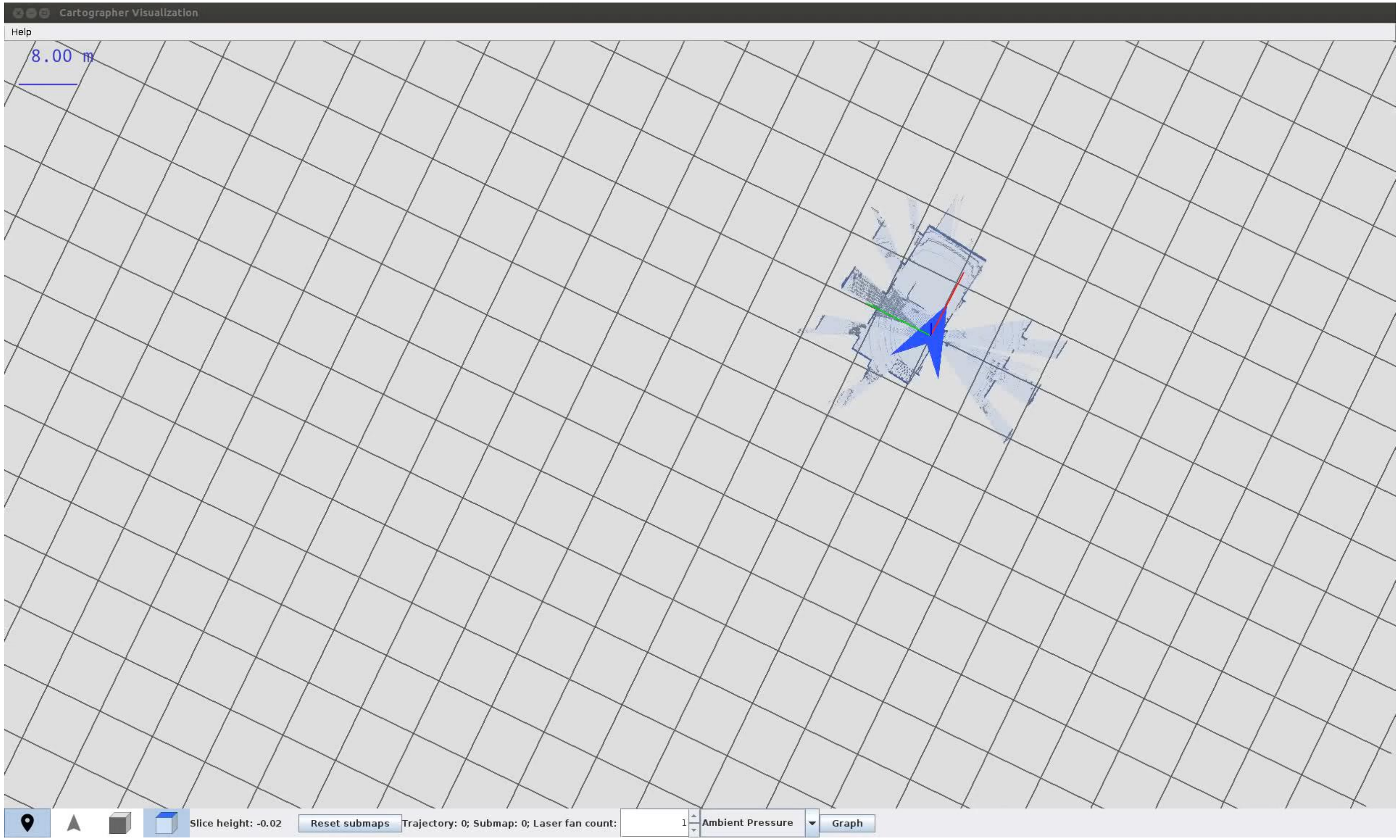
Examples of SLAM systems



Source Code: https://github.com/erik-nelson/blam

Examples of SLAM systems

Google Cartographer: 2D and 3D laser SLAM



Code: https://github.com/googlecartographer/cartographer

SLAM: possible problem definitions

- Smoothing/Batch/Full SLAM

$$p(\mathbf{x}_{1:T}, \mathbf{m} \mid \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1}, \mathbf{x}_0)$$

Initial state

Sequence of states

Map

Sequence of sensor measurements

Sequence of commands

SLAM: possible problem definitions

- Smoothing/Batch/Full SLAM

Initial state

$$p(\mathbf{x}_{1:T}, \mathbf{m} \mid \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1}, \mathbf{x}_0)$$

Sequence of states

Map

Sequence of sensor measurements

Sequence of commands

- Filtering SLAM

$$p(\mathbf{x}_t, \mathbf{m}_t \mid \mathbf{z}_{0:t}, \mathbf{u}_{0:t-1}, \mathbf{x}_0)$$

SLAM: possible problem definitions

In this lecture

- Smoothing/Batch/Full SLAM

Initial state

$$p(\mathbf{x}_{1:T}, \mathbf{m} \mid \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1}, \mathbf{x}_0)$$

Sequence of states

Map

Sequence of sensor measurements

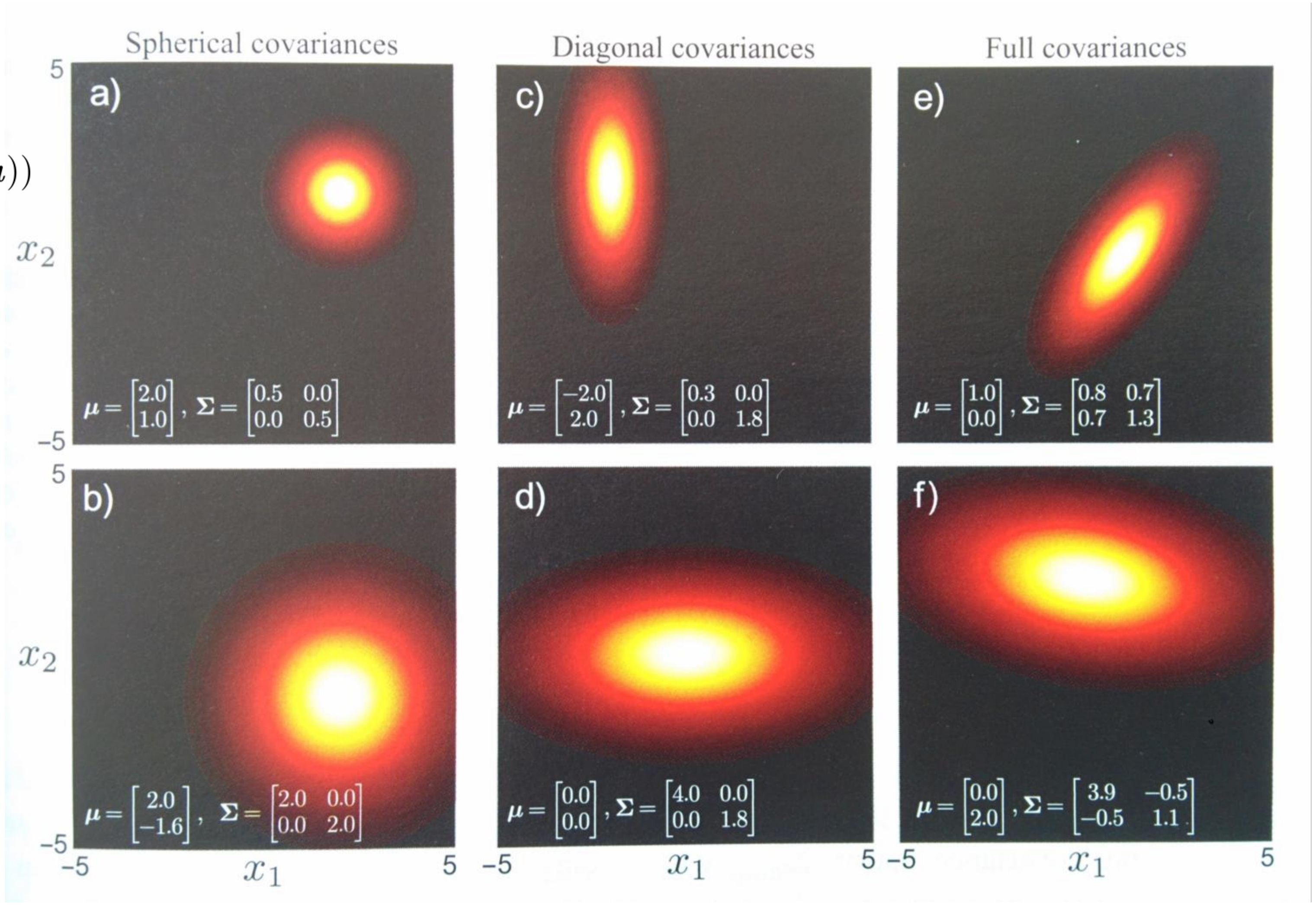
Sequence of commands

- Filtering SLAM

$$p(\mathbf{x}_t, \mathbf{m}_t \mid \mathbf{z}_{0:t}, \mathbf{u}_{0:t-1}, \mathbf{x}_0)$$

Background: Multivariate Gaussian Distribution

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}\det(\boldsymbol{\Sigma})^{1/2}}\exp(-0.5(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}))$$



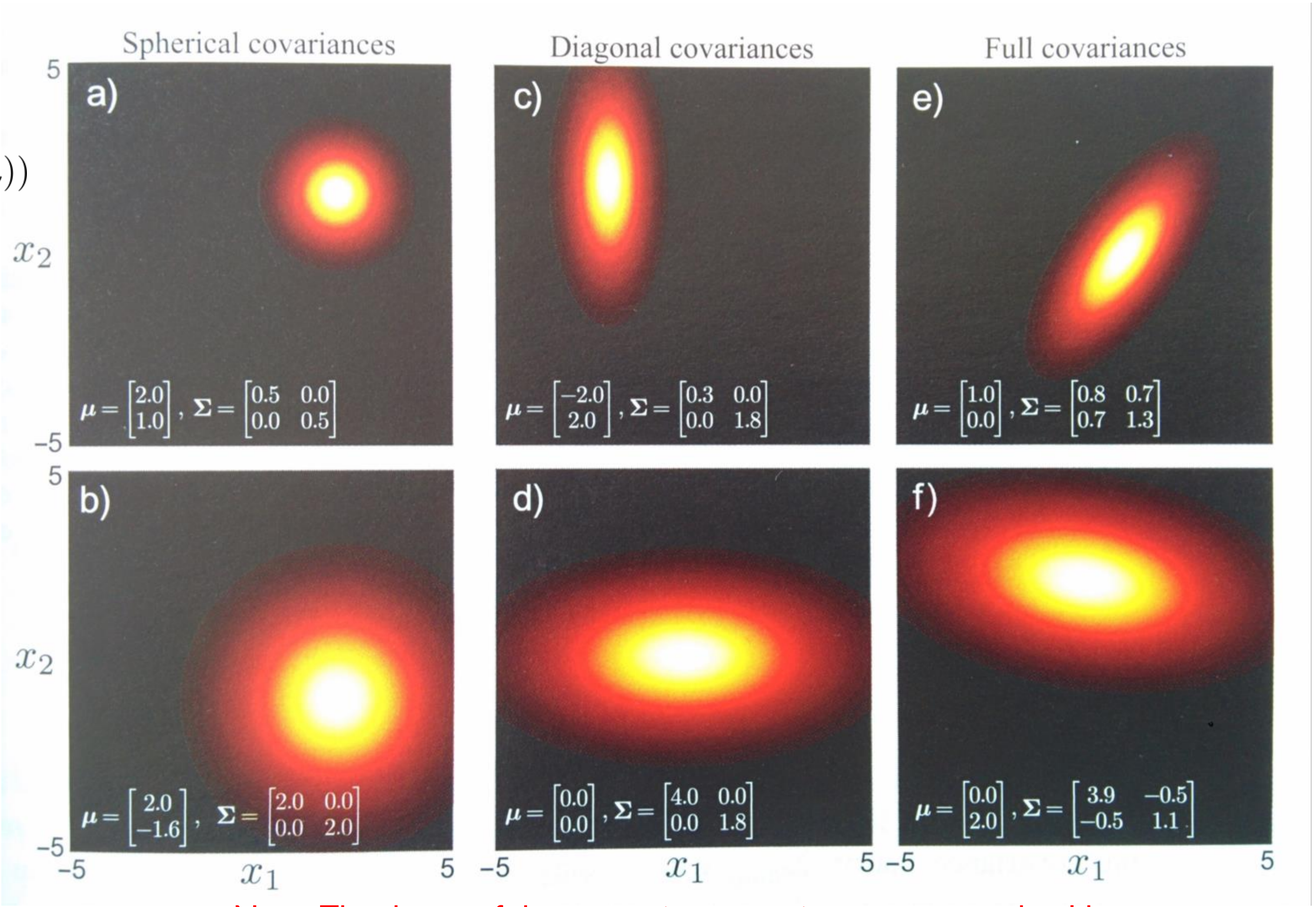
From "Computer Vision: Models, Learning, and Inference" Simon Prince

Background: Multivariate Gaussian Distribution

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}\det(\mathbf{\Sigma})^{1/2}}\exp(-0.5(\mathbf{x}-\boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}))$$

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}\det(\mathbf{\Sigma})^{1/2}}\exp(-0.5||\mathbf{x}-\boldsymbol{\mu}||^2_{\mathbf{\Sigma}})$$

Shortcut notation: $||\mathbf{x}||^2_{\mathbf{\Sigma}} = \mathbf{x}^T\mathbf{\Sigma}^{-1}\mathbf{x}$



From "Computer Vision: Models, Learning, and Inference" Simon Prince

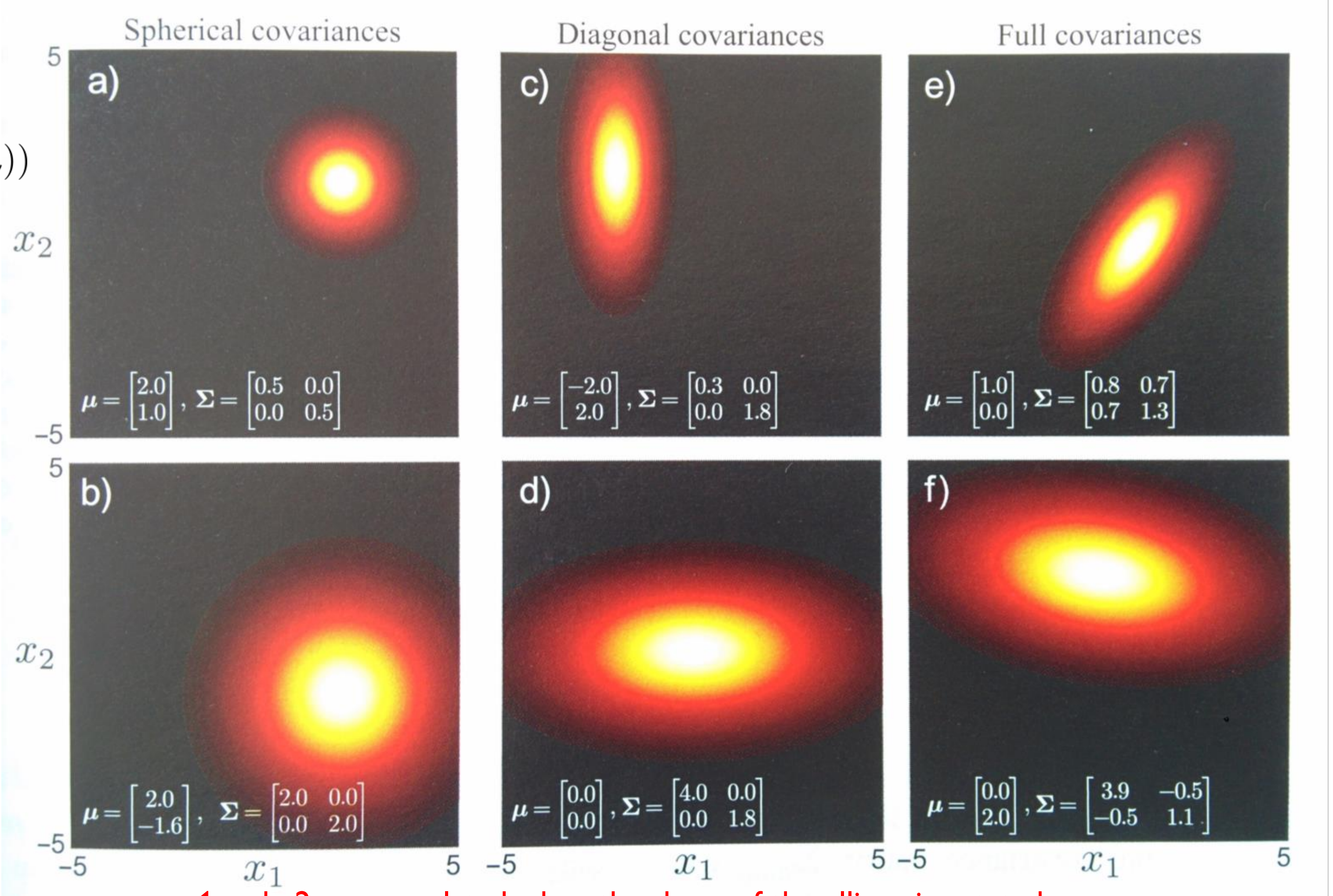
Note: The shapes of these covariances are important, you should know them well. In particular, when are x1 and x2 correlated?

Background: Multivariate Gaussian Distribution

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}\det(\boldsymbol{\Sigma})^{1/2}}\exp(-0.5(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}))$$

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}\det(\boldsymbol{\Sigma})^{1/2}}\exp(-0.5||\mathbf{x}-\boldsymbol{\mu}||^2_{\boldsymbol{\Sigma}})$$

Shortcut notation: $||\mathbf{x}||^2_{\boldsymbol{\Sigma}} = \mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x}$



x1 and x2 are correlated when the shape of the ellipse is rotated,
i.e. when there are nonzero off-diagonal terms in the covariance matrix.
In this example, (e) and (f)

From “Computer Vision: Models, Learning, and Inference” Simon Prince

Confidence regions

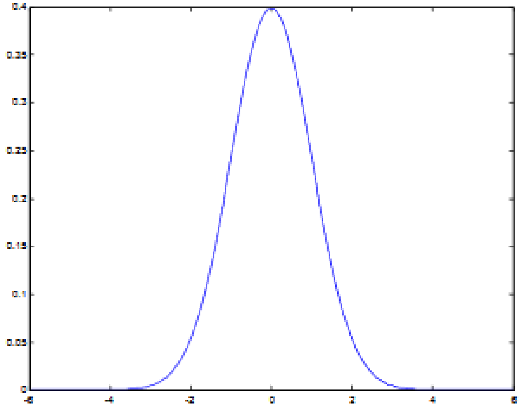
- To quantify confidence and uncertainty define a confidence region R about a point x (e.g. the mode) such that at a confidence level $c \leq 1$

$$p(x \in R) = c$$

- we can then say (for example) there is a 99% probability that the true value is in R

- e.g. for a univariate normal distribution $N(\mu,\sigma^2)$

$$\begin{aligned} p(|x-\mu| < \sigma) &\approx 0.67 \\ p(|x-\mu| < 2\sigma) &\approx 0.95 \\ p(|x-\mu| < 3\sigma) &\approx 0.997 \end{aligned}$$



Expectation

- Expected value of a random variable X:

$$\mathbb{E}_{x\sim p(X)}[X] = \int_x xp(X=x)dx$$

- E is linear: $\mathbb{E}_{x\sim p(X)}[X+c] = \mathbb{E}_{x\sim p(X)}[X] + c$

$$\mathbb{E}_{x\sim p(X)}[AX+b] = A\mathbb{E}_{x\sim p(X)}[X] + b$$

- If X,Y are independent then [Note: inverse does not hold]

$$\mathbb{E}_{x,y\sim p(X,Y)}[XY] = \mathbb{E}_{x\sim p(X)}[X]\,\mathbb{E}_{x\sim p(Y)}[Y]$$

Covariance Matrix

- Measures linear dependence between random variables X, Y. Does **not** measure independence.

$$\mathrm{Cov}[X, Y] = E[XY] - E[X]E[Y]$$

- Variance of X

$$\mathrm{Var}[X] = \mathrm{Cov}[X] = \mathrm{Cov}[X, X] = E[X^2] - E[X]^2$$

$$\mathrm{Cov}[AX + b] = A\mathrm{Cov}[X]A^T$$

$$\mathrm{Cov}[X + Y] = \mathrm{Cov}[X] + \mathrm{Cov}[Y] - 2\mathrm{Cov}[X, Y]$$

Covariance Matrix

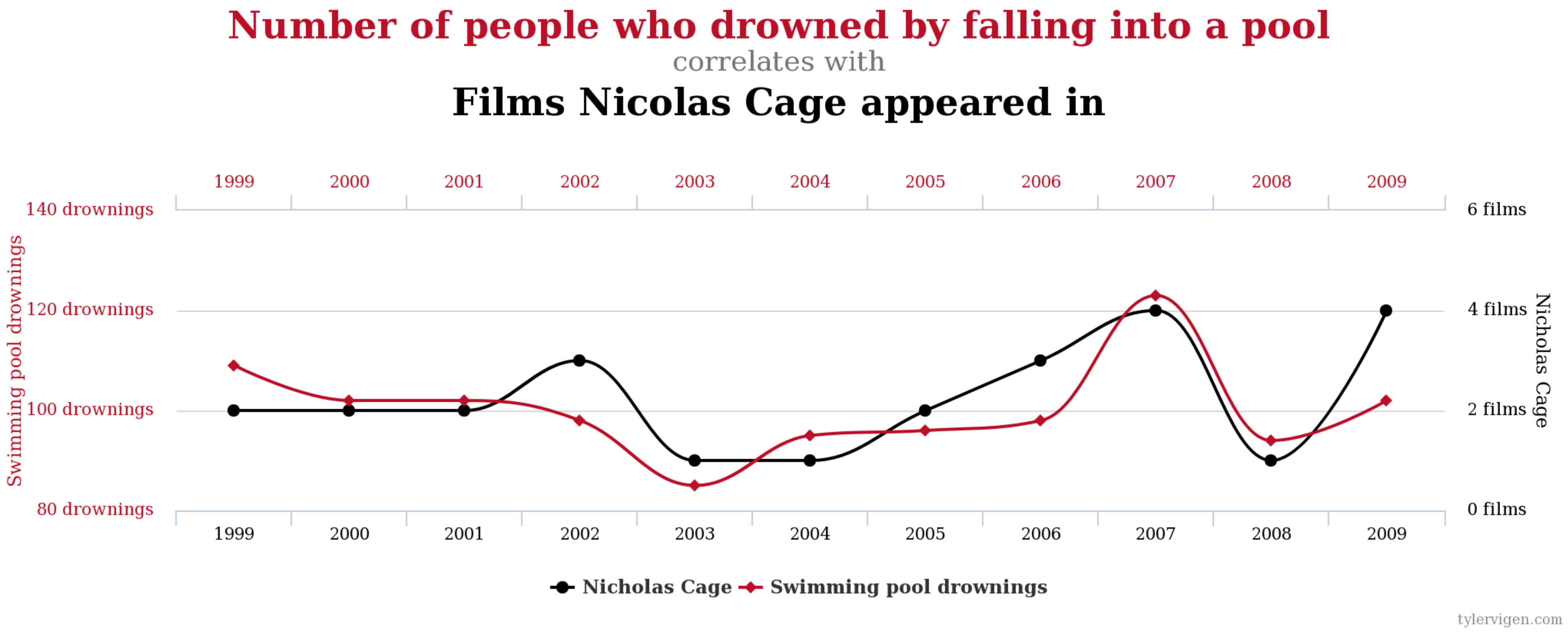
- Measures linear dependence between random variables X, Y. Does **not** measure independence.

$$\mathrm{Cov}[X,Y] = E[XY] - E[X]E[Y]$$

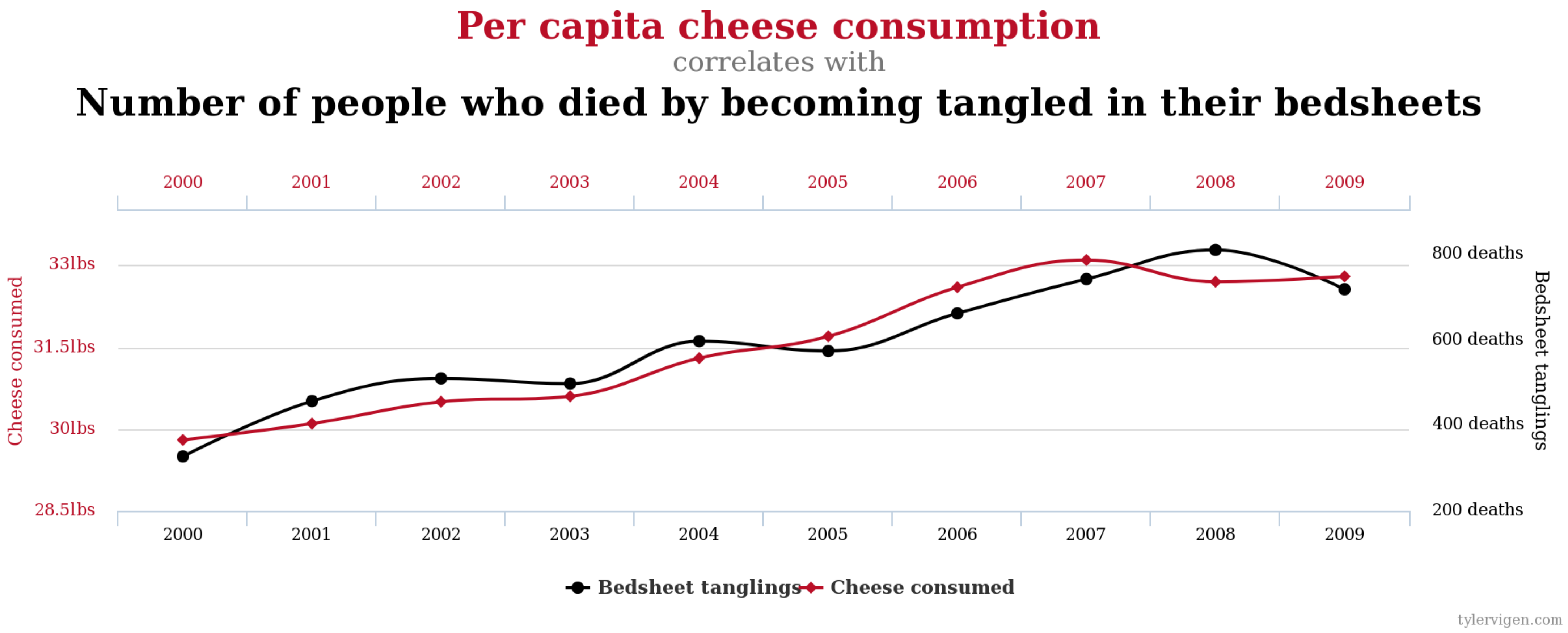
- Entry (i,j) of the covariance matrix measures whether changes in variable X_i co-occur with changes in variable Y_j

- It does not measure whether one causes the other.

Correlation does not imply causation



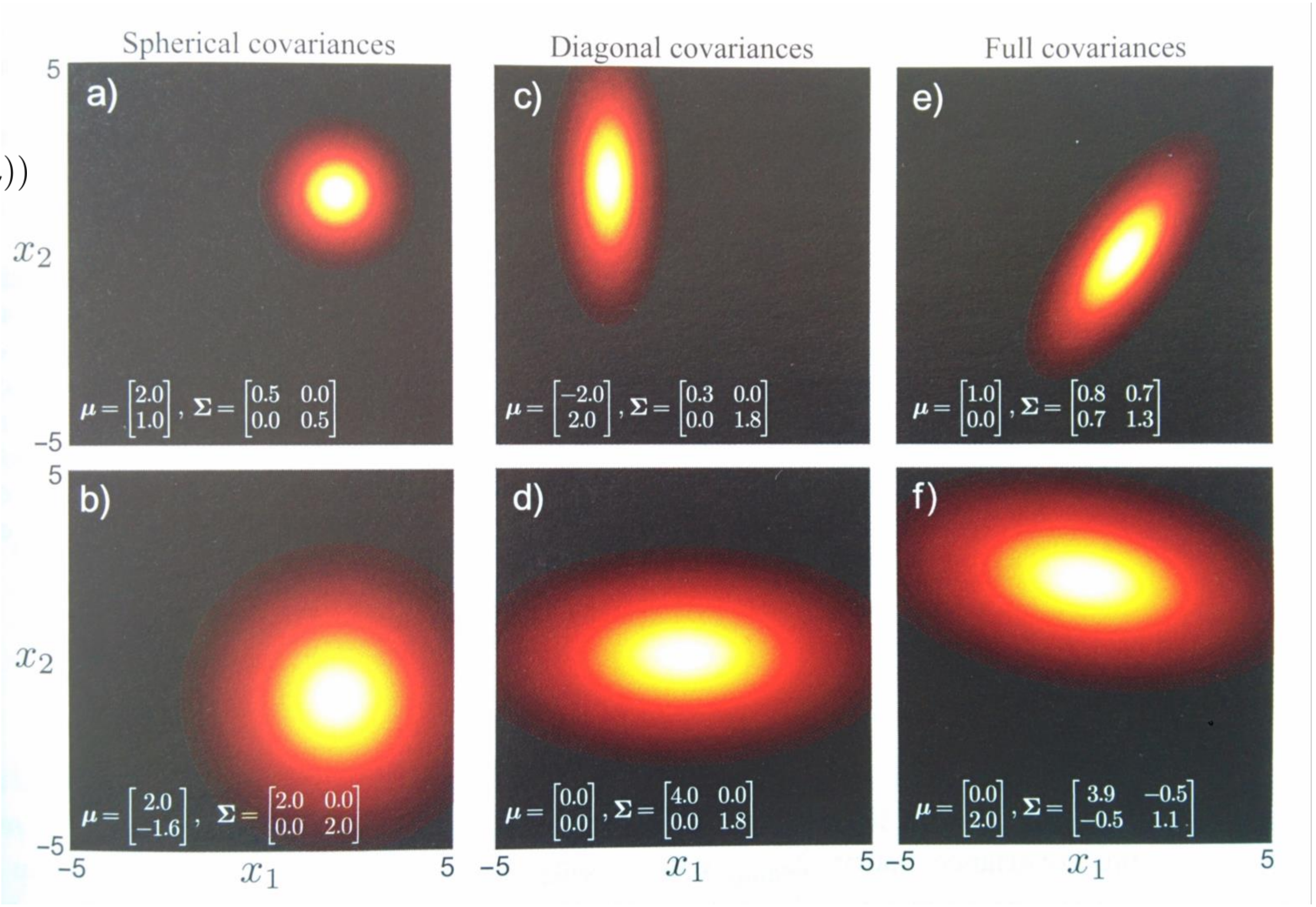
Correlation does not imply causation



Background: Multivariate Gaussian Distribution

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}\det(\boldsymbol{\Sigma})^{1/2}}\exp(-0.5(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}))$$

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}\det(\boldsymbol{\Sigma})^{1/2}}\exp(-0.5||\mathbf{x}-\boldsymbol{\mu}||^2_{\boldsymbol{\Sigma}})$$



For multivariate Gaussians:

$$E[\mathbf{x}] = \boldsymbol{\mu}$$

$$\mathrm{Cov}[\mathbf{x}] = \boldsymbol{\Sigma}$$

From "Computer Vision: Models, Learning, and Inference" Simon Prince

Background: Multivariate Gaussian Distribution

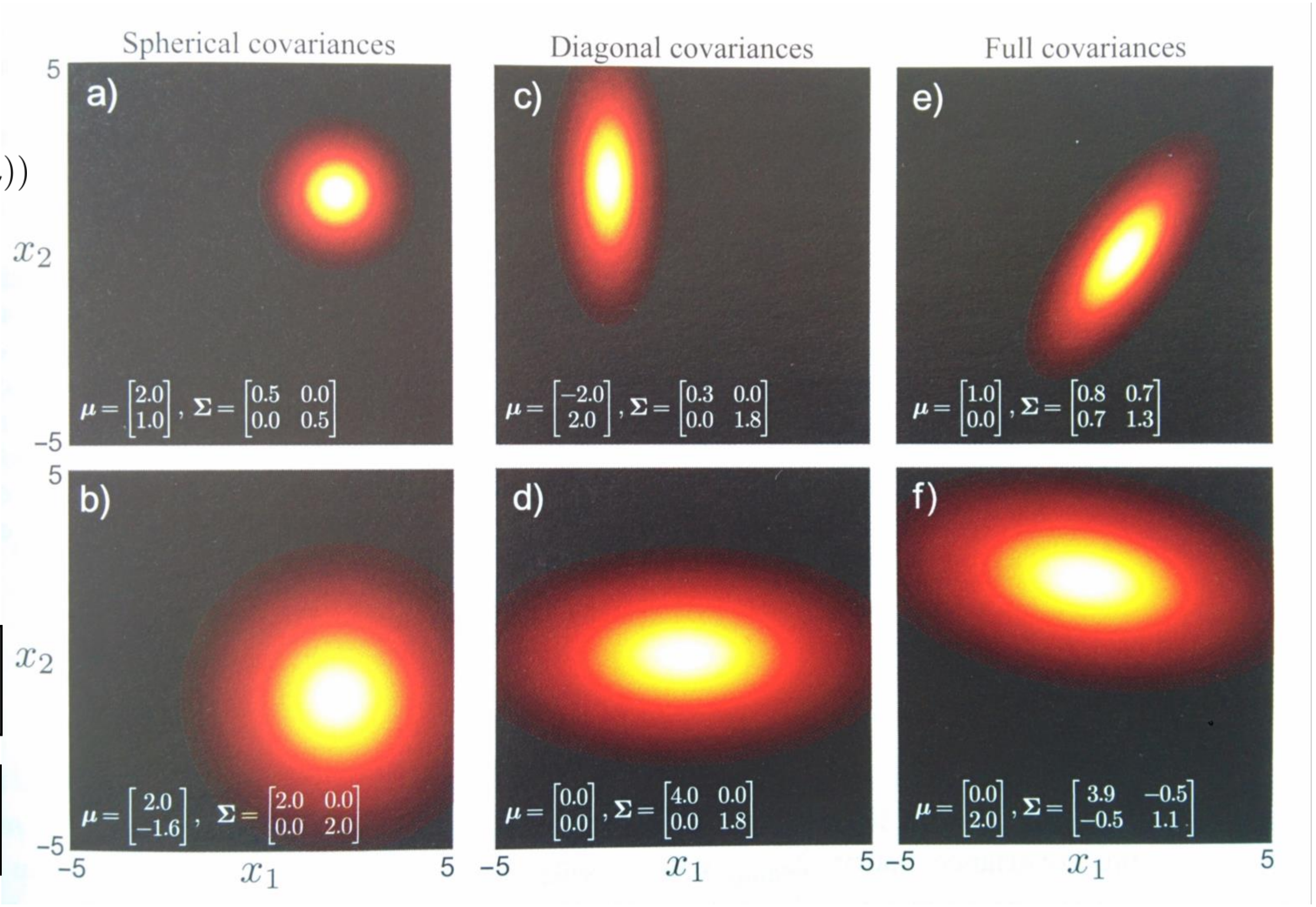
$$p(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}\det(\mathbf{\Sigma})^{1/2}}\exp(-0.5(\mathbf{x}-\boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}))$$

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}\det(\mathbf{\Sigma})^{1/2}}\exp(-0.5||\mathbf{x}-\boldsymbol{\mu}||^2_{\mathbf{\Sigma}})$$

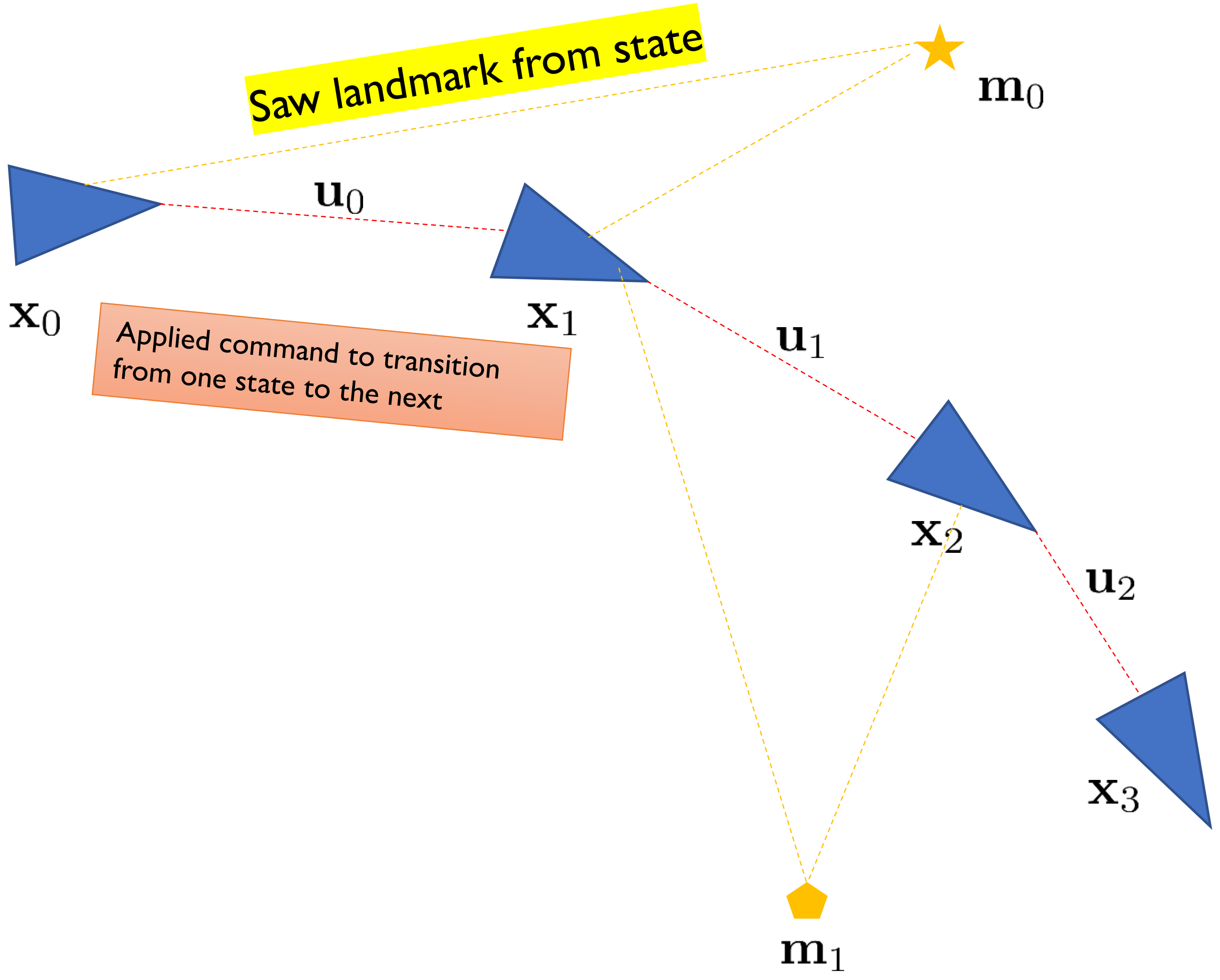
Since we have 2D examples here:

$$\mathrm{Cov}[\mathbf{x}] = \mathbf{\Sigma} = \begin{bmatrix}\mathrm{Cov}[x_1,x_1] & \mathrm{Cov}[x_1,x_2]\\ \mathrm{Cov}[x_2,x_1] & \mathrm{Cov}[x_2,x_2]\end{bmatrix}$$

$$= \begin{bmatrix}\mathrm{Var}[x_1] & \mathrm{Cov}[x_1,x_2]\\ \mathrm{Cov}[x_2,x_1] & \mathrm{Var}[x_2]\end{bmatrix}$$



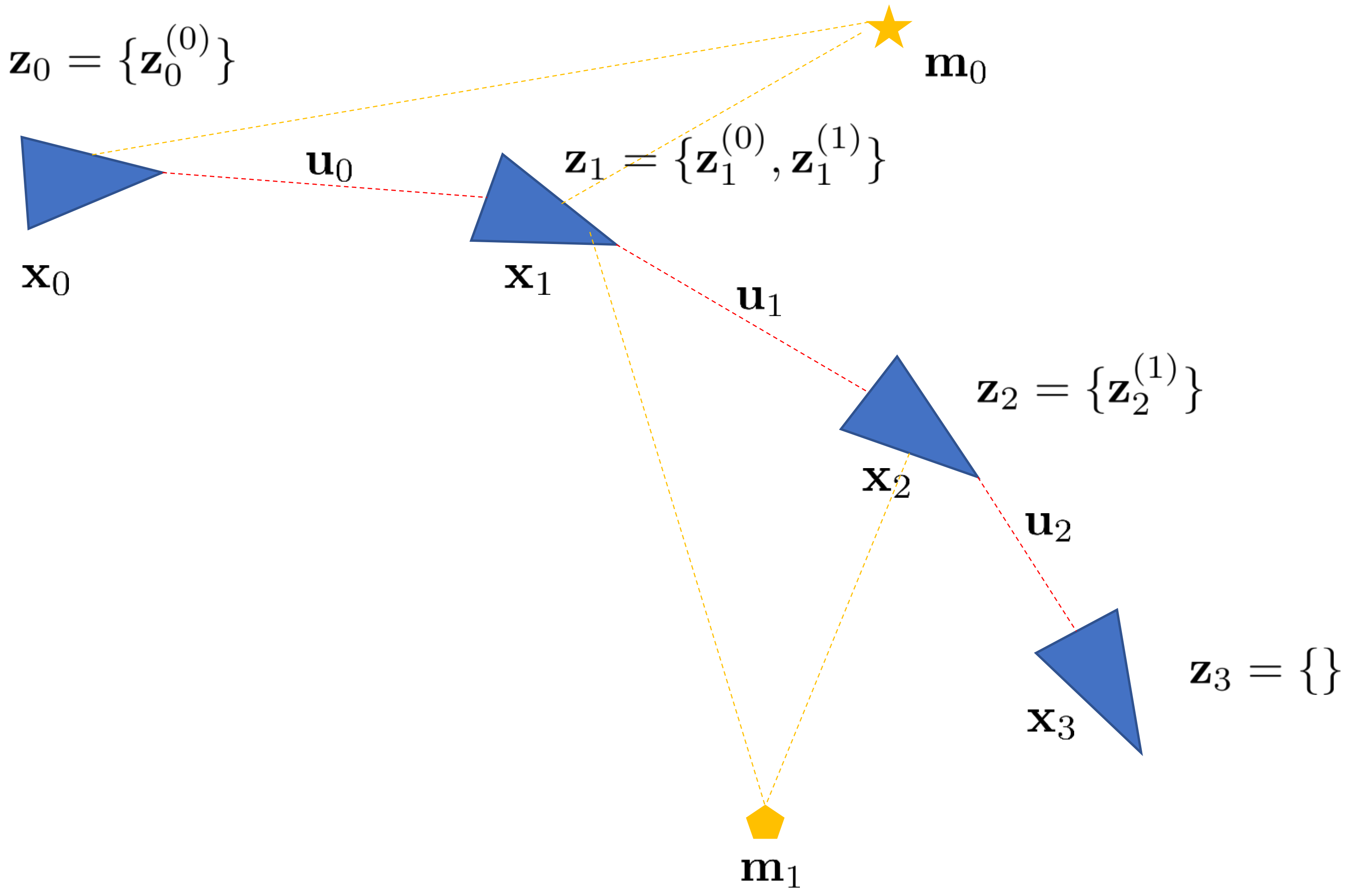
SLAM: graph representation



Map $\mathbf{m} = \{\mathbf{m}_0, \mathbf{m}_1\}$ consists of landmarks that are easily identifiable and cannot be mistaken for one another.

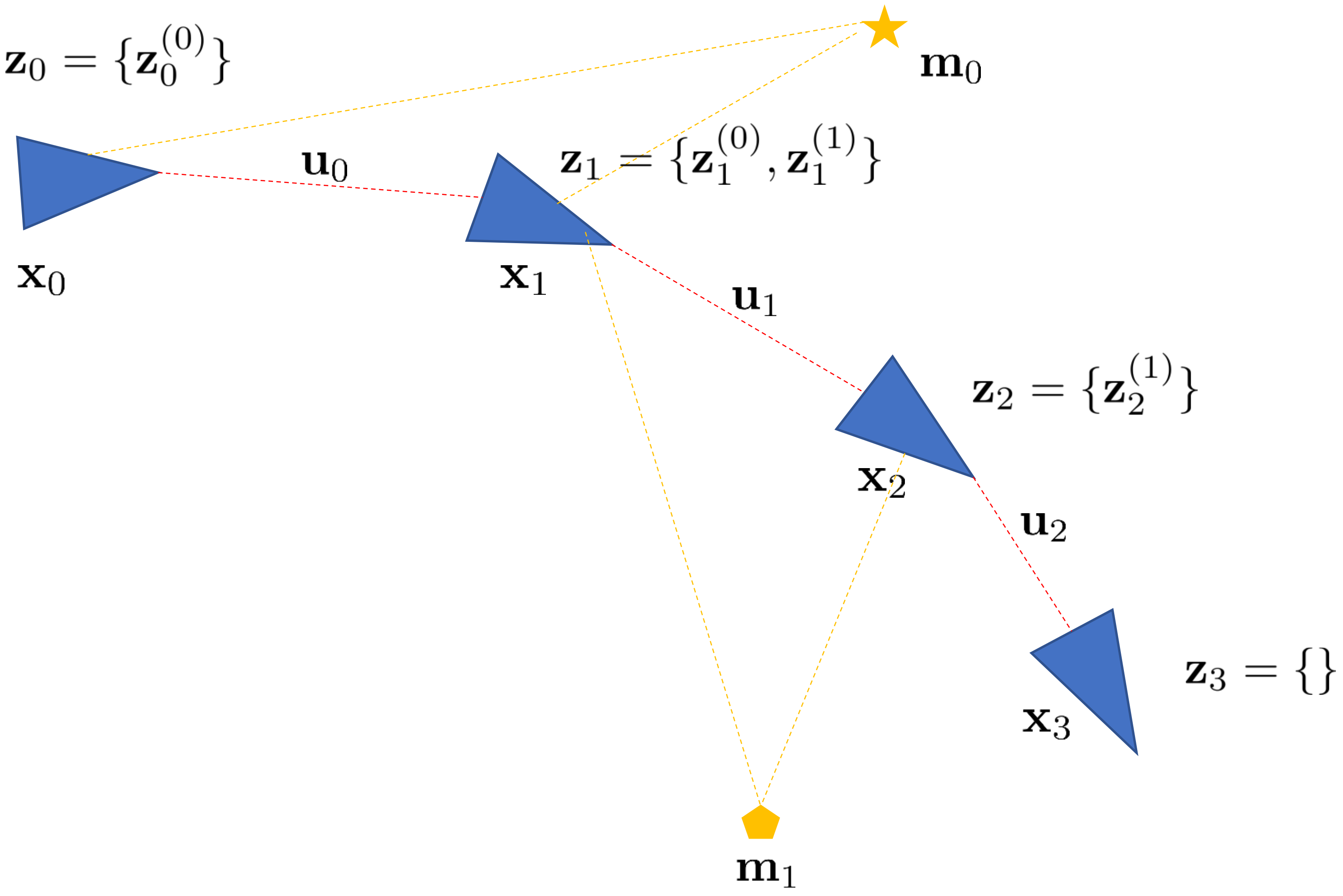
i.e. we are avoiding the data association problem here.

SLAM: graph representation



Map $\mathbf{m} = \{\mathbf{m}_0, \mathbf{m}_1\}$ consists of landmarks that are easily identifiable and cannot be mistaken for one another.

SLAM: graph representation



Notice that the graph is mostly sparse as long as not many states observe the same landmark.

That implies that there are many symbolic dependencies between random variables in

$$p(\mathbf{x}_{1:T}, \mathbf{m} \mid \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1}, \mathbf{x}_0)$$

that are not necessary and can be dropped.

GraphSLAM: SLAM as a Maximum A Posteriori Estimate

Instead of computing the posterior $p(\mathbf{x}_{1:T}, \mathbf{m} \mid \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1}, \mathbf{x}_0)$ we are going to compute its max

See least squares lecture

$$\mathbf{x}^*_{1:T}, \mathbf{m}^* = \underset{\mathbf{x}_{1:T}, \mathbf{m}}{\operatorname{argmax}}\, p(\mathbf{x}_{1:T}, \mathbf{m} \mid \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1}, \mathbf{x}_0)$$

GraphSLAM:
SLAM as a Maximum A Posteriori Estimate

Instead of computing the posterior $p(\mathbf{x}_{1:T}, \mathbf{m} \mid \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1}, \mathbf{x}_0)$ we are going to compute its max

$$\begin{aligned}
\mathbf{x}^*_{1:T}, \mathbf{m}^* &= \underset{\mathbf{x}_{1:T}, \mathbf{m}}{\operatorname{argmax}}\; p(\mathbf{x}_{1:T}, \mathbf{m} \mid \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1}, \mathbf{x}_0) \\
&= \underset{\mathbf{x}_{1:T}, \mathbf{m}}{\operatorname{argmax}} \left[\frac{p(\mathbf{x}_{1:T}, \mathbf{m}, \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1} | \mathbf{x}_0)}{p(\mathbf{z}_{0:T}, \mathbf{u}_{0:T-1} | \mathbf{x}_0)} \right]
\end{aligned}$$

by definition of conditional distribution

GraphSLAM: SLAM as a Maximum A Posteriori Estimate

Instead of computing the posterior $p(\mathbf{x}_{1:T}, \mathbf{m} \mid \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1}, \mathbf{x}_0)$ we are going to compute its max

$$
\begin{aligned}
\mathbf{x}^*_{1:T}, \mathbf{m}^* &= \underset{\mathbf{x}_{1:T}, \mathbf{m}}{\operatorname{argmax}}\, p(\mathbf{x}_{1:T}, \mathbf{m} \mid \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1}, \mathbf{x}_0) \\
&= \underset{\mathbf{x}_{1:T}, \mathbf{m}}{\operatorname{argmax}} \left[\frac{p(\mathbf{x}_{1:T}, \mathbf{m}, \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1} | \mathbf{x}_0)}{p(\mathbf{z}_{0:T}, \mathbf{u}_{0:T-1} | \mathbf{x}_0)} \right] \\
&= \underset{\mathbf{x}_{1:T}, \mathbf{m}}{\operatorname{argmax}}\, p(\mathbf{x}_{1:T}, \mathbf{m}, \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1} | \mathbf{x}_0)
\end{aligned}
$$

denominator does not depend on optimization variables

GraphSLAM:
SLAM as a Maximum A Posteriori Estimate

Instead of computing the posterior $p(\mathbf{x}_{1:T}, \mathbf{m} \mid \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1}, \mathbf{x}_0)$ we are going to compute its max

$$
\begin{aligned}
\mathbf{x}^*_{1:T}, \mathbf{m}^* &= \underset{\mathbf{x}_{1:T}, \mathbf{m}}{\operatorname{argmax}}\, p(\mathbf{x}_{1:T}, \mathbf{m} \mid \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1}, \mathbf{x}_0) \\
&= \underset{\mathbf{x}_{1:T}, \mathbf{m}}{\operatorname{argmax}} \left[\frac{p(\mathbf{x}_{1:T}, \mathbf{m}, \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1} | \mathbf{x}_0)}{p(\mathbf{z}_{0:T}, \mathbf{u}_{0:T-1} | \mathbf{x}_0)} \right] \\
&= \underset{\mathbf{x}_{1:T}, \mathbf{m}}{\operatorname{argmax}}\, p(\mathbf{x}_{1:T}, \mathbf{m}, \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1} | \mathbf{x}_0) \\
&= \underset{\mathbf{x}_{1:T}, \mathbf{m}}{\operatorname{argmax}} \left[\prod_{t=1}^{T} p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_{t-1}) \prod_{t=0}^{T} \prod_{\mathbf{z}_t^{(k)} \in \mathbf{z}_t} p(\mathbf{z}_t^{(k)} | \mathbf{x}_t, \mathbf{m}_k) \right]
\end{aligned}
$$

See Appendix 1 for the derivation of this step

Observation of landmark k at time t

Set of observations that were made at time t

Set of observations that were made at time t

GraphSLAM: SLAM as a Maximum A Posteriori Estimate

Instead of computing the posterior $p(\mathbf{x}_{1:T}, \mathbf{m} \mid \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1}, \mathbf{x}_0)$ we are going to compute its max

$$
\begin{aligned}
\mathbf{x}_{1:T}^*, \mathbf{m}^* &= \underset{\mathbf{x}_{1:T}, \mathbf{m}}{\operatorname{argmax}}\, p(\mathbf{x}_{1:T}, \mathbf{m} \mid \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1}, \mathbf{x}_0) \\
&= \underset{\mathbf{x}_{1:T}, \mathbf{m}}{\operatorname{argmax}} \left[\frac{p(\mathbf{x}_{1:T}, \mathbf{m}, \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1} | \mathbf{x}_0)}{p(\mathbf{z}_{0:T}, \mathbf{u}_{0:T-1} | \mathbf{x}_0)} \right] \\
&= \underset{\mathbf{x}_{1:T}, \mathbf{m}}{\operatorname{argmax}}\, p(\mathbf{x}_{1:T}, \mathbf{m}, \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1} | \mathbf{x}_0) \\
&= \underset{\mathbf{x}_{1:T}, \mathbf{m}}{\operatorname{argmax}} \left[\prod_{t=1}^{T} \underbrace{p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_{t-1})}_{\text{Probabilistic dynamics model}} \prod_{t=0}^{T} \prod_{\mathbf{z}_t^{(k)} \in \mathbf{z}_t} \underbrace{p(\mathbf{z}_t^{(k)} | \mathbf{x}_t, \mathbf{m}_k)}_{\text{Probabilistic sensor measurement model}} \right]
\end{aligned}
$$

GraphSLAM: SLAM as a Maximum A Posteriori Estimate

Instead of computing the posterior $p(\mathbf{x}_{1:T}, \mathbf{m} \mid \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1}, \mathbf{x}_0)$ we are going to compute its max

$$
\begin{aligned}
\mathbf{x}_{1:T}^*, \mathbf{m}^* &= \underset{\mathbf{x}_{1:T}, \mathbf{m}}{\operatorname{argmax}}\, p(\mathbf{x}_{1:T}, \mathbf{m} \mid \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1}, \mathbf{x}_0) \\
&= \underset{\mathbf{x}_{1:T}, \mathbf{m}}{\operatorname{argmax}} \left[\frac{p(\mathbf{x}_{1:T}, \mathbf{m}, \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1} | \mathbf{x}_0)}{p(\mathbf{z}_{0:T}, \mathbf{u}_{0:T-1} | \mathbf{x}_0)} \right] \\
&= \underset{\mathbf{x}_{1:T}, \mathbf{m}}{\operatorname{argmax}}\, p(\mathbf{x}_{1:T}, \mathbf{m}, \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1} | \mathbf{x}_0) \\
&= \underset{\mathbf{x}_{1:T}, \mathbf{m}}{\operatorname{argmax}} \left[\prod_{t=1}^{T} p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_{t-1}) \prod_{t=0}^{T} \prod_{\mathbf{z}_t^{(k)} \in \mathbf{z}_t} p(\mathbf{z}_t^{(k)} | \mathbf{x}_t, \mathbf{m}_k) \right] \\
&= \underset{\mathbf{x}_{1:T}, \mathbf{m}}{\operatorname{argmax}} \left[\sum_{t=1}^{T} \log p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_{t-1}) + \sum_{t=0}^{T} \sum_{\mathbf{z}_t^{(k)} \in \mathbf{z}_t} \log p(\mathbf{z}_t^{(k)} | \mathbf{x}_t, \mathbf{m}_k) \right]
\end{aligned}
$$

GraphSLAM:
SLAM as a Maximum A Posteriori Estimate

Instead of computing the posterior $p(\mathbf{x}_{1:T}, \mathbf{m} \mid \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1}, \mathbf{x}_0)$ we are going to compute its max

$$\mathbf{x}^*_{1:T}, \mathbf{m}^* = \underset{\mathbf{x}_{1:T}, \mathbf{m}}{\operatorname{argmax}} \left[\sum_{t=1}^{T} \log p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_{t-1}) + \sum_{t=0}^{T} \sum_{\mathbf{z}_t^{(k)} \in \mathbf{z}_t} \log p(\mathbf{z}_t^{(k)} | \mathbf{x}_t, \mathbf{m}_k) \right]$$

Main GraphSLAM assumptions:

1. Uncertainty in the dynamics model is Gaussian

$$\mathbf{x}_t = \mathbf{f}(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}) + \mathbf{w}_t$$
$$\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_t)$$

so

$$\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_{t-1} \sim \mathcal{N}(\mathbf{f}(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}), \mathbf{R}_t)$$

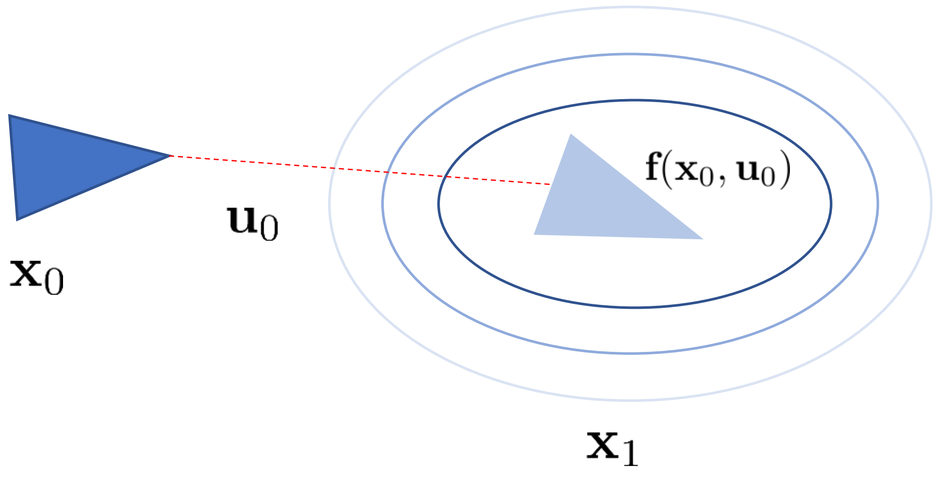
2. Uncertainty in the sensor model is Gaussian

$$\mathbf{z}_t^{(k)} = \mathbf{h}(\mathbf{x}_t, \mathbf{m}_k) + \mathbf{v}_t$$
$$\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_t)$$

so

$$\mathbf{z}_t^{(k)} | \mathbf{x}_t, \mathbf{m}_k \sim \mathcal{N}(\mathbf{h}(\mathbf{x}_t, \mathbf{m}_k), \mathbf{Q}_t)$$

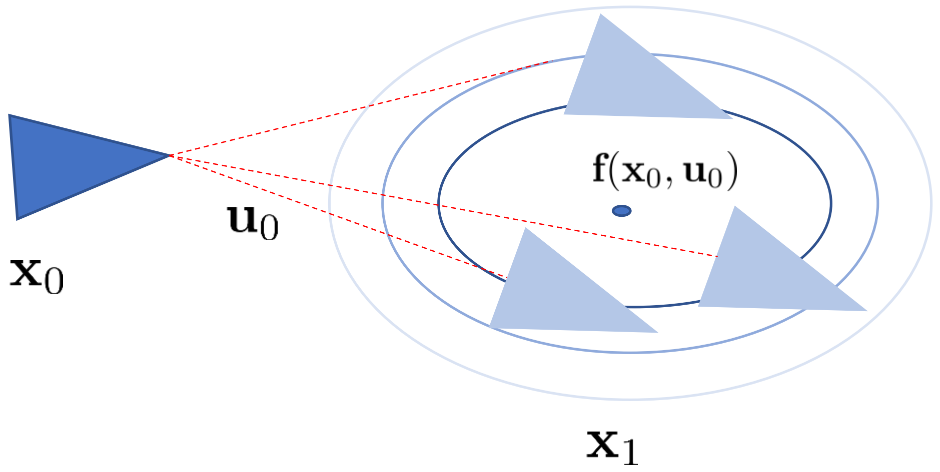
SLAM: noise/errors



$$\mathbf{x}_1 | \mathbf{x}_0, \mathbf{u}_0 \sim \mathcal{N}(\mathbf{f}(\mathbf{x}_0, \mathbf{u}_0), \mathbf{R}_0)$$

Expected to end up at $\mathbf{x}_1 = \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0)$ from $\mathbf{x}_0$

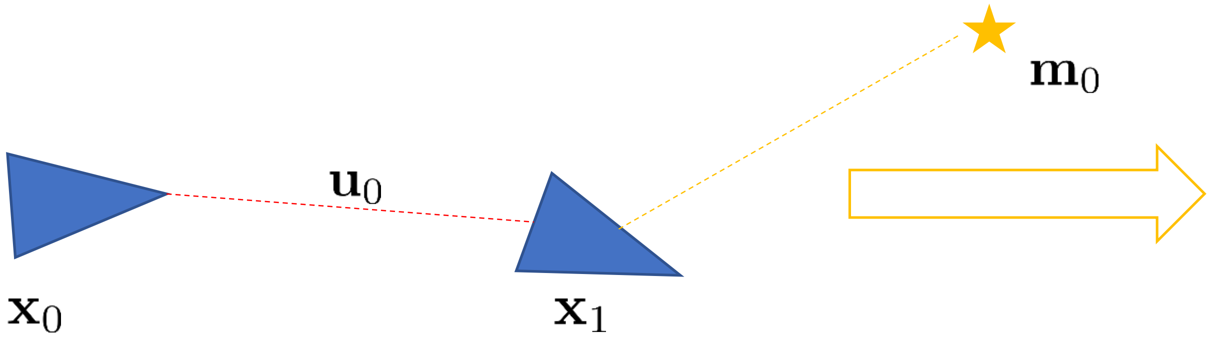
SLAM: noise/errors

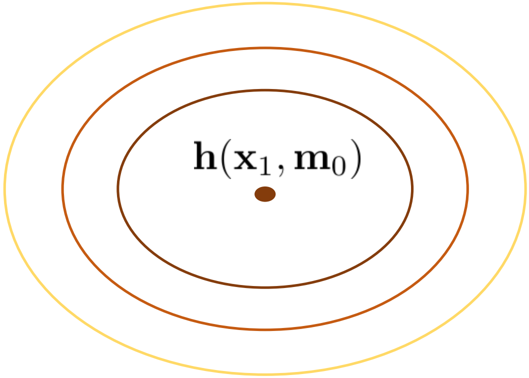


$$\mathbf{x}_1 | \mathbf{x}_0, \mathbf{u}_0 \sim \mathcal{N}(\mathbf{f}(\mathbf{x}_0, \mathbf{u}_0), \mathbf{R}_0)$$

Expected to end up at $\mathbf{x}_1 = \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0)$ from $\mathbf{x}_0$ but we might end up around it, within the ellipse defined by the covariance matrix $\mathbf{R}_0$

SLAM: noise/errors





$$\mathbf{z}_1^{(0)} | \mathbf{x}_1, \mathbf{m}_0 \sim \mathcal{N}(\mathbf{h}(\mathbf{x}_1, \mathbf{m}_0), \mathbf{Q}_1)$$

Expected to get measurement $\mathbf{h}(\mathbf{x}_1, \mathbf{m}_0)$ at state $\mathbf{x}_1$ but it might be somewhere within the ellipse defined by the covariance matrix $\mathbf{Q}_1$

GraphSLAM: SLAM as a least squares problem

Instead of computing the posterior $p(\mathbf{x}_{1:T}, \mathbf{m} \mid \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1}, \mathbf{x}_0)$ we are going to compute its max

$$\begin{aligned}
\mathbf{x}^*_{1:T}, \mathbf{m}^* &= \underset{\mathbf{x}_{1:T}, \mathbf{m}}{\text{argmax}} \left[\sum_{t=1}^{T} \log p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_{t-1}) + \sum_{t=0}^{T} \sum_{\mathbf{z}_t^{(k)} \in \mathbf{z}_t} \log p(\mathbf{z}_t^{(k)} | \mathbf{x}_t, \mathbf{m}_k) \right] \\
&= \underset{\mathbf{x}_{1:T}, \mathbf{m}}{\text{argmax}} \left[-\sum_{t=1}^{T} ||\mathbf{x}_t - \mathbf{f}(\mathbf{x}_{t-1}, \mathbf{u}_{t-1})||^2_{\mathbf{R}_t} - \sum_{t=0}^{T} \sum_{\mathbf{z}_t^{(k)} \in \mathbf{z}_t} ||\mathbf{z}_t^{(k)} - \mathbf{h}(\mathbf{x}_t, \mathbf{m}_k)||^2_{\mathbf{Q}_t} \right]
\end{aligned}$$

Notation:

$\mathbf{x}^T \mathbf{Q}^{-1} \mathbf{x} = ||\mathbf{x}||^2_{\mathbf{Q}}$

$\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_{t-1} \sim \mathcal{N}(\mathbf{f}(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}), \mathbf{R}_t)$ $\quad$ $\mathbf{z}_t^{(k)} | \mathbf{x}_t, \mathbf{m}_k \sim \mathcal{N}(\mathbf{h}(\mathbf{x}_t, \mathbf{m}_k), \mathbf{Q}_t)$

GraphSLAM: SLAM as a least squares problem

Instead of computing the posterior $p(\mathbf{x}_{1:T}, \mathbf{m} \mid \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1}, \mathbf{x}_0)$ we are going to compute its max

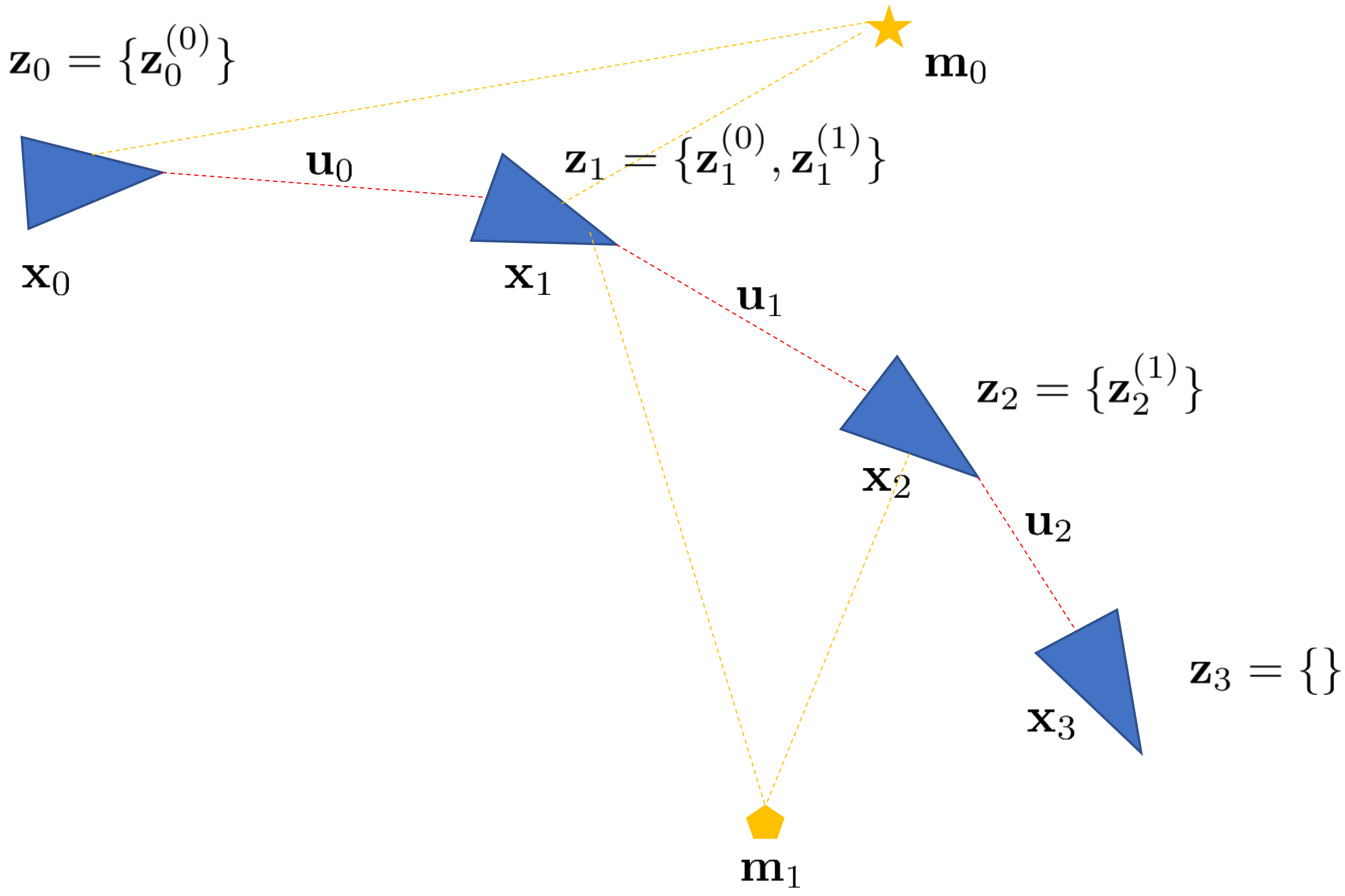
$$
\begin{aligned}
\mathbf{x}^*_{1:T}, \mathbf{m}^* &= \underset{\mathbf{x}_{1:T},\mathbf{m}}{\text{argmax}} \left[\sum_{t=1}^{T} \log p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_{t-1}) + \sum_{t=0}^{T} \sum_{\mathbf{z}_t^{(k)} \in \mathbf{z}_t} \log p(\mathbf{z}_t^{(k)} | \mathbf{x}_t, \mathbf{m}_k) \right] \\
&= \underset{\mathbf{x}_{1:T},\mathbf{m}}{\text{argmax}} \left[-\sum_{t=1}^{T} ||\mathbf{x}_t - \mathbf{f}(\mathbf{x}_{t-1}, \mathbf{u}_{t-1})||^2_{\mathbf{R}_t} - \sum_{t=0}^{T} \sum_{\mathbf{z}_t^{(k)} \in \mathbf{z}_t} ||\mathbf{z}_t^{(k)} - \mathbf{h}(\mathbf{x}_t, \mathbf{m}_k)||^2_{\mathbf{Q}_t} \right] \\
&= \underset{\mathbf{x}_{1:T},\mathbf{m}}{\text{argmin}} \left[\sum_{t=1}^{T} ||\mathbf{x}_t - \mathbf{f}(\mathbf{x}_{t-1}, \mathbf{u}_{t-1})||^2_{\mathbf{R}_t} + \sum_{t=0}^{T} \sum_{\mathbf{z}_t^{(k)} \in \mathbf{z}_t} ||\mathbf{z}_t^{(k)} - \mathbf{h}(\mathbf{x}_t, \mathbf{m}_k)||^2_{\mathbf{Q}_t} \right]
\end{aligned}
$$

Notation:

$\mathbf{x}^T \mathbf{Q}^{-1} \mathbf{x} = ||\mathbf{x}||^2_{\mathbf{Q}}$

$\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_{t-1} \sim \mathcal{N}(\mathbf{f}(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}), \mathbf{R}_t)$ $\quad$ $\mathbf{z}_t^{(k)} | \mathbf{x}_t, \mathbf{m}_k \sim \mathcal{N}(\mathbf{h}(\mathbf{x}_t, \mathbf{m}_k), \mathbf{Q}_t)$

GraphSLAM: example



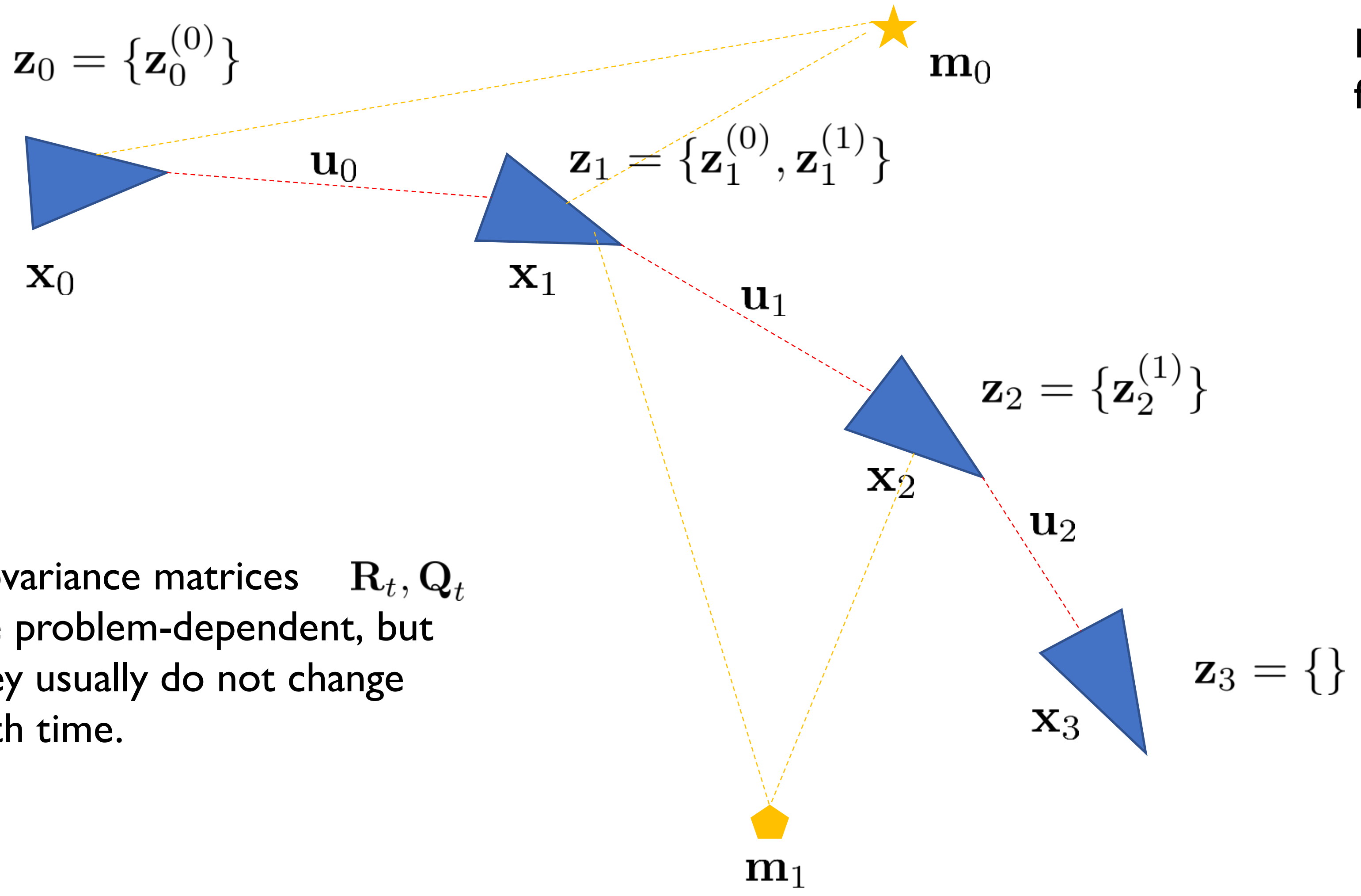
Need to minimize the sum of the following quadratic terms:

$$||\mathbf{x}_1 - \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0)||^2_{\mathbf{R}_1}$$
$$||\mathbf{x}_2 - \mathbf{f}(\mathbf{x}_1, \mathbf{u}_1)||^2_{\mathbf{R}_2}$$
$$||\mathbf{x}_3 - \mathbf{f}(\mathbf{x}_2, \mathbf{u}_2)||^2_{\mathbf{R}_3}$$
$$||\mathbf{z}_0^{(0)} - \mathbf{h}(\mathbf{x}_0, \mathbf{m}_0)||^2_{\mathbf{Q}_0}$$
$$||\mathbf{z}_1^{(0)} - \mathbf{h}(\mathbf{x}_1, \mathbf{m}_0)||^2_{\mathbf{Q}_1}$$
$$||\mathbf{z}_1^{(1)} - \mathbf{h}(\mathbf{x}_1, \mathbf{m}_1)||^2_{\mathbf{Q}_1}$$
$$||\mathbf{z}_2^{(1)} - \mathbf{h}(\mathbf{x}_2, \mathbf{m}_1)||^2_{\mathbf{Q}_2}$$

with respect to variables:
$\mathbf{x}_1 \quad \mathbf{x}_2 \quad \mathbf{x}_3 \quad \mathbf{m}_0 \quad \mathbf{m}_1$

initial state $\mathbf{x}_0$ is given

GraphSLAM: example



Covariance matrices $\mathbf{R}_t, \mathbf{Q}_t$ are problem-dependent, but they usually do not change with time.

Need to minimize the sum of the following quadratic terms:

$$||\mathbf{x}_1 - \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0)||^2_{\mathbf{R}_1}$$
$$||\mathbf{x}_2 - \mathbf{f}(\mathbf{x}_1, \mathbf{u}_1)||^2_{\mathbf{R}_2}$$
$$||\mathbf{x}_3 - \mathbf{f}(\mathbf{x}_2, \mathbf{u}_2)||^2_{\mathbf{R}_3}$$
$$||\mathbf{z}_0^{(0)} - \mathbf{h}(\mathbf{x}_0, \mathbf{m}_0)||^2_{\mathbf{Q}_0}$$
$$||\mathbf{z}_1^{(0)} - \mathbf{h}(\mathbf{x}_1, \mathbf{m}_0)||^2_{\mathbf{Q}_1}$$
$$||\mathbf{z}_1^{(1)} - \mathbf{h}(\mathbf{x}_1, \mathbf{m}_1)||^2_{\mathbf{Q}_1}$$
$$||\mathbf{z}_2^{(1)} - \mathbf{h}(\mathbf{x}_2, \mathbf{m}_1)||^2_{\mathbf{Q}_2}$$

with respect to variables:
$\mathbf{x}_1 \quad \mathbf{x}_2 \quad \mathbf{x}_3 \quad \mathbf{m}_0 \quad \mathbf{m}_1$

initial state $\mathbf{x}_0$ is given

Examples of dynamics and sensor models

$$\mathbf{x}_t = \mathbf{f}(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}) + \mathbf{w}_t$$

Can be any of the dynamical systems we saw in Lecture 2.

$$\mathbf{z}_t^{(k)} = \mathbf{h}(\mathbf{x}_t, \mathbf{m}_k) + \mathbf{v}_t$$

$\mathbf{z}_t^{(k)}$ can be any of the sensors we saw in Lecture 4:

- Laser scan $\{(r_i, \theta_i)\}_{i=1:K}$ where $\mathbf{m}_k$ is an occupancy grid
- Range and bearing (r, θ) to the landmark $\mathbf{m}_k = (x_k, y_k, z_k)$
- Bearing measurements from images
- Altitude/Depth
- Gyroscope
- Accelerometer

Appendix 1

Claim: $$p(\mathbf{x}_{1:T}, \mathbf{m}, \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1}, \mathbf{x}_0) \quad = \quad p(\mathbf{x}_0) \prod_{t=1}^{T} p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_{t-1}) \prod_{t=0}^{T} \prod_{\mathbf{z}_t^{(k)} \in \, \mathbf{z}_t} p(\mathbf{z}_t^{(k)} | \mathbf{x}_t, \mathbf{m}_k)$$

Proof:

$$\begin{aligned}
p(\mathbf{x}_{1:T}, \mathbf{m}, \mathbf{z}_{0:T}, \mathbf{u}_{0:T-1}, \mathbf{x}_0) \quad &= \quad p(\mathbf{z}_T | \mathbf{x}_{1:T}, \mathbf{m}, \mathbf{z}_{0:T-1}, \mathbf{u}_{0:T-1}, \mathbf{x}_0) \; p(\mathbf{x}_{1:T}, \mathbf{m}, \mathbf{z}_{0:T-1}, \mathbf{u}_{0:T-1}, \mathbf{x}_0) \\
&= \quad p(\mathbf{z}_T | \mathbf{x}_T, \mathbf{m}) \; p(\mathbf{x}_{1:T}, \mathbf{m}, \mathbf{z}_{0:T-1}, \mathbf{u}_{0:T-1}, \mathbf{x}_0) \\
&= \quad p(\mathbf{z}_T | \mathbf{x}_T, \mathbf{m}) \; p(\mathbf{z}_{T-1} | \mathbf{x}_{1:T}, \mathbf{m}, \mathbf{z}_{0:T-2}, \mathbf{u}_{0:T-1}, \mathbf{x}_0) \; p(\mathbf{x}_{1:T}, \mathbf{m}, \mathbf{z}_{0:T-2}, \mathbf{u}_{0:T-1}, \mathbf{x}_0) \\
&= \quad p(\mathbf{z}_T | \mathbf{x}_T, \mathbf{m}) \; p(\mathbf{z}_{T-1} | \mathbf{x}_{T-1}, \mathbf{m}) \; p(\mathbf{x}_{1:T}, \mathbf{m}, \mathbf{z}_{0:T-2}, \mathbf{u}_{0:T-1}, \mathbf{x}_0) \\
&\ldots \\
&= \quad \prod_{t=0}^{T} p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m}) \; p(\mathbf{x}_{1:T}, \mathbf{m}, \mathbf{u}_{0:T-1}, \mathbf{x}_0) \\
&= \quad \prod_{t=0}^{T} p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m}) \; p(\mathbf{x}_T | \mathbf{x}_{1:T-1}, \mathbf{m}, \mathbf{u}_{0:T-1}, \mathbf{x}_0) \;\; p(\mathbf{x}_{1:T-1}, \mathbf{m}, \mathbf{u}_{0:T-1}, \mathbf{x}_0) \\
&= \quad \prod_{t=0}^{T} p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m}) \; p(\mathbf{x}_T | \mathbf{x}_{T-1}, \mathbf{u}_{T-1}) \; p(\mathbf{x}_{1:T-1}, \mathbf{m}, \mathbf{u}_{0:T-1}, \mathbf{x}_0) \\
&= \quad \prod_{t=0}^{T} p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m}) \; p(\mathbf{x}_T | \mathbf{x}_{T-1}, \mathbf{u}_{T-1}) \; p(\mathbf{x}_{T-1} | \mathbf{x}_{T-2}, \mathbf{m}, \mathbf{u}_{0:T-1}, \mathbf{x}_0) \; p(\mathbf{x}_{1:T-2}, \mathbf{m}, \mathbf{u}_{0:T-1}, \mathbf{x}_0) \\
&\ldots \\
&= \quad \left[\prod_{t=0}^{T} p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m}) \right] \; p(\mathbf{x}_0) \; \prod_{t=1}^{T} p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_{t-1})
\end{aligned}$$

Appendix 1

Claim: $$p(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m}) = \prod_{\mathbf{z}_t^{(k)} \in \mathbf{z}_t} p(\mathbf{z}_t^{(k)}|\mathbf{x}_t,\mathbf{m}_k)$$ where $\mathbf{z}_t = \{\mathbf{z}_t^{(k)} \text{ iff landmark } \mathbf{m}_k \text{ was observed}\}$
$\mathbf{m} = \{\text{landmarks } \mathbf{m}_k\}$

Proof:

Suppose without loss of generality that $\mathbf{z}_t = \{\mathbf{z}_t^{(k)},\ k = 1...K\}$ and $\mathbf{m} = \{\mathbf{m}_k,\ k = 1...K\}$
i.e. that all landmarks were observed from the state at time t. Then:

$$\begin{aligned}
p(\mathbf{z}_t^{(1)},...,\mathbf{z}_t^{(K)}|\mathbf{x}_t,\mathbf{m}) &= p(\mathbf{z}_t^{(1)}|\mathbf{z}_t^{(2)},...,\mathbf{z}_t^{(K)},\mathbf{x}_t,\mathbf{m})\ p(\mathbf{z}_t^{(2)},...,\mathbf{z}_t^{(K)}|\mathbf{x}_t,\mathbf{m}) \\
&= p(\mathbf{z}_t^{(1)}|\mathbf{x}_t,\mathbf{m}_1)\ p(\mathbf{z}_t^{(2)},...,\mathbf{z}_t^{(K)}|\mathbf{x}_t,\mathbf{m}) \\
&= p(\mathbf{z}_t^{(1)}|\mathbf{x}_t,\mathbf{m}_1)\ p(\mathbf{z}_t^{(2)}|\mathbf{z}_t^{(3)},...,\mathbf{z}_t^{(K)},\mathbf{x}_t,\mathbf{m})\ p(\mathbf{z}_t^{(3)},...,\mathbf{z}_t^{(K)}|\mathbf{x}_t,\mathbf{m}) \\
&= p(\mathbf{z}_t^{(1)}|\mathbf{x}_t,\mathbf{m}_1)\ p(\mathbf{z}_t^{(2)}|\mathbf{x}_t,\mathbf{m}_2)\ p(\mathbf{z}_t^{(3)},...,\mathbf{z}_t^{(K)}|\mathbf{x}_t,\mathbf{m}) \\
&\ldots \\
&= \prod_{k=1}^{K} p(\mathbf{z}_t^{(k)}|\mathbf{x}_t,\mathbf{m}_k)
\end{aligned}$$