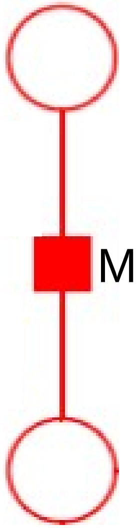
Lecture 5: Learning (cont)

Probabilistic Graphical Models, Koller and Friedman:

- Chap 19

- MAP Inference on the SLAM Graph
- Partially Observed Data, Gradient Ascent, Expectation Maximization, Gaussian Mixture Learning

SLAM Factor Graph



Feature

Measurement

Pose

Where are the Parameters?

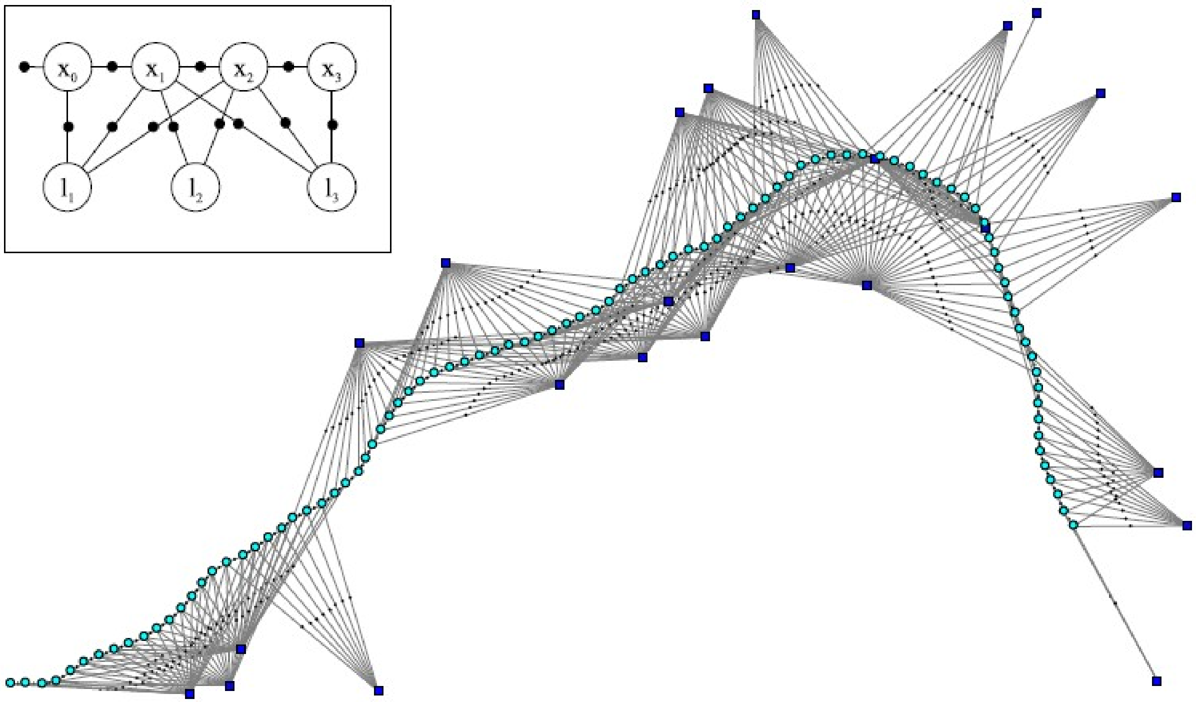
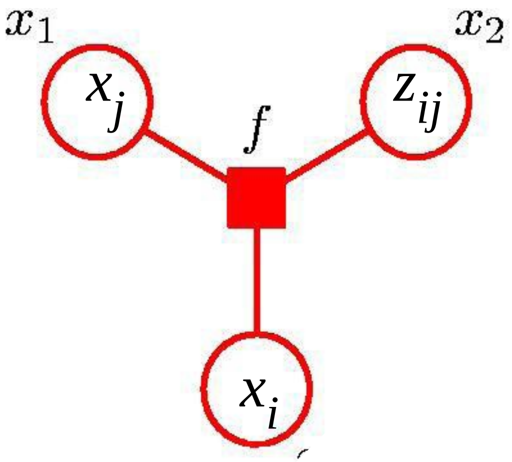


Figure from Dellart and Kaess ‘Square Root SAM: Simultaneous localization and mapping via square root information smoothing’.

Feature

Measurement



Pose

MLE Estimate on SLAM Graph

The factor graph is actually a conditional factor graph

- $p(D \mid \boldsymbol{\theta}) = p(\{\mathbf{z}_{\mathrm{ij}}\} \mid \{\boldsymbol{x}_i\})$ is what we will maximize to find the $\boldsymbol{x}$'s, so we are doing MLE!
- But: $p(\{\boldsymbol{x}_i\} \mid \{\mathbf{z}_{\mathbf{ij}}\}) = p(\{\mathbf{z}_{\mathrm{ij}}\} \mid \{\boldsymbol{x}_i\})\, p(\{\boldsymbol{x}_i\}) / p(\{\mathbf{z}_{\mathrm{ij}}\})$
- So if all paths and maps are equally likely (uniform prior) then MAP would be the same.

MLE Estimate on SLAM Graph

- SLAM graphs (normally) have Gaussian Factors that connect two nodes, (with an implied known measurement).
- $p(D \mid \boldsymbol{\theta})=p(\{\boldsymbol{z}_{\mathrm{ij}}\} \mid \{\boldsymbol{x}_{\boldsymbol{i}}\})$

$$\propto |\Omega|^{1/2}\exp[-½ \textstyle\sum_{cij} (\boldsymbol{h}_{\mathrm{ij}}(\boldsymbol{x}_{\mathrm{i}}, \boldsymbol{x}_{\mathrm{j}})-\boldsymbol{z}_{\mathrm{ij}})^{\mathrm{T}}\, \Omega \mathrm{ij}\ (\boldsymbol{h}_{\mathrm{ij}}(\boldsymbol{x}_{\mathrm{i}}, \boldsymbol{x}_{\mathrm{j}})-\boldsymbol{z}_{\mathrm{ij}})]$$

$$l(\boldsymbol{x})= ½ \textstyle\sum_{cij} (\boldsymbol{h}_{\mathrm{ij}}(\boldsymbol{x}_{\mathrm{i}}, \boldsymbol{x}_{\mathrm{j}})-\boldsymbol{z}_{\mathrm{ij}})^{\mathrm{T}}\, \Omega \mathrm{ij}\ (\boldsymbol{h}_{\mathrm{ij}}(\boldsymbol{x}_{\mathrm{i}}, \boldsymbol{x}_{\mathrm{j}})-\boldsymbol{z}_{\mathrm{ij}})-½ \ln|\Omega|$$

- $l(\boldsymbol{x})\approx ½ \sum_{cij} \|A_{\mathrm{ij}}(\boldsymbol{x}_{\mathrm{i}} - \bar{\boldsymbol{x}}_{\mathrm{i}}, \boldsymbol{x}_{\mathrm{j}} - \bar{\boldsymbol{x}}_{\boldsymbol{j}})-\boldsymbol{b}_{\mathrm{ij}}\|^{2}$ -(constants)

MAP Inference on SLAM Graph

- SLAM graphs (normally) have Gaussian Factors that connect two nodes.
- Nodes are either feature or pose nodes.

- The -log-likelihood then looks like:

$$l(\boldsymbol{x}, \boldsymbol{c}) = \tfrac{1}{2} \sum_{cij} (\boldsymbol{h}_{\mathrm{ij}}(\boldsymbol{x}_{\mathrm{i}}, \boldsymbol{x}_{\mathrm{j}}) - \boldsymbol{z}_{\mathrm{ij}})^{\mathrm{T}}\, \Omega\, (\boldsymbol{h}_{\mathrm{ij}}(\boldsymbol{x}_{\mathrm{i}}, \boldsymbol{x}_{\mathrm{j}}) - \boldsymbol{z}_{\mathrm{ij}})$$

c just indicates the choice of where to put edges.

- Linearize and take a square root of the matrix in the middle:

$$l(\boldsymbol{x}, \boldsymbol{c}) = \tfrac{1}{2} \sum_{cij} \|A_{\mathrm{ij}}(\boldsymbol{x}_{\mathrm{i}} - \bar{\boldsymbol{x}}_{\mathrm{i}}, \boldsymbol{x}_{\mathrm{j}} - \bar{\boldsymbol{x}}_{\boldsymbol{j}}) - \boldsymbol{b}_{\mathrm{ij}}\|^{2}$$

$$A^{T} A\, \boldsymbol{\Delta x} = A^{T} \boldsymbol{b}$$

MAP Inference on SLAM Graph

- SLAM graphs (normally) have Gaussian Factors that connect two nodes.
- Nodes are either feature or pose nodes.

- The -log-likelihood then looks like:

$$l(\boldsymbol{x}, \boldsymbol{c}) = \tfrac{1}{2} \sum_{cij} (\boldsymbol{h}_{ij}(\boldsymbol{x}_i, \boldsymbol{x}_j) - \boldsymbol{z}_{ij})^T \, \Omega \, (\boldsymbol{h}_{ij}(\boldsymbol{x}_i, \boldsymbol{x}_j) - \boldsymbol{z}_{ij})$$

c just indicates the choice of where to put edges.

What?

- Linearize and take a square root of the matrix in the middle:

$$l(\boldsymbol{x}, \boldsymbol{c}) = \tfrac{1}{2} \sum_{cij} \| A_{ij}(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i, \boldsymbol{x}_j - \bar{\boldsymbol{x}}_{\boldsymbol{j}}) - \boldsymbol{b}_{ij} \|^2$$

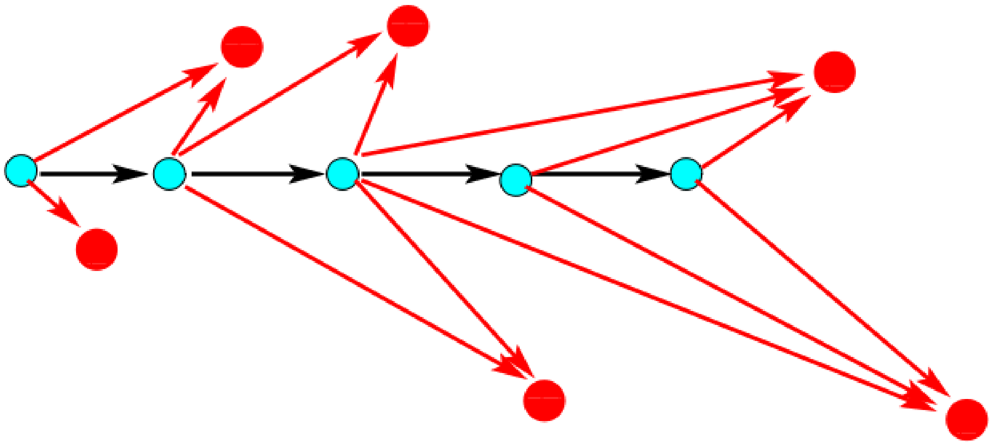
$$A^T A \, \boldsymbol{\Delta x} = A^T \boldsymbol{b}$$

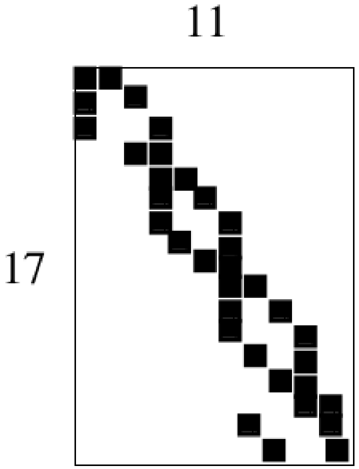
Sparsity

🔴 = Features 🔵 = Robot Poses

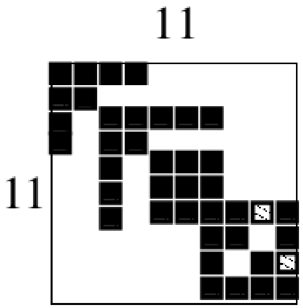
→ = Feature Measurements

→ = Motion Measurements





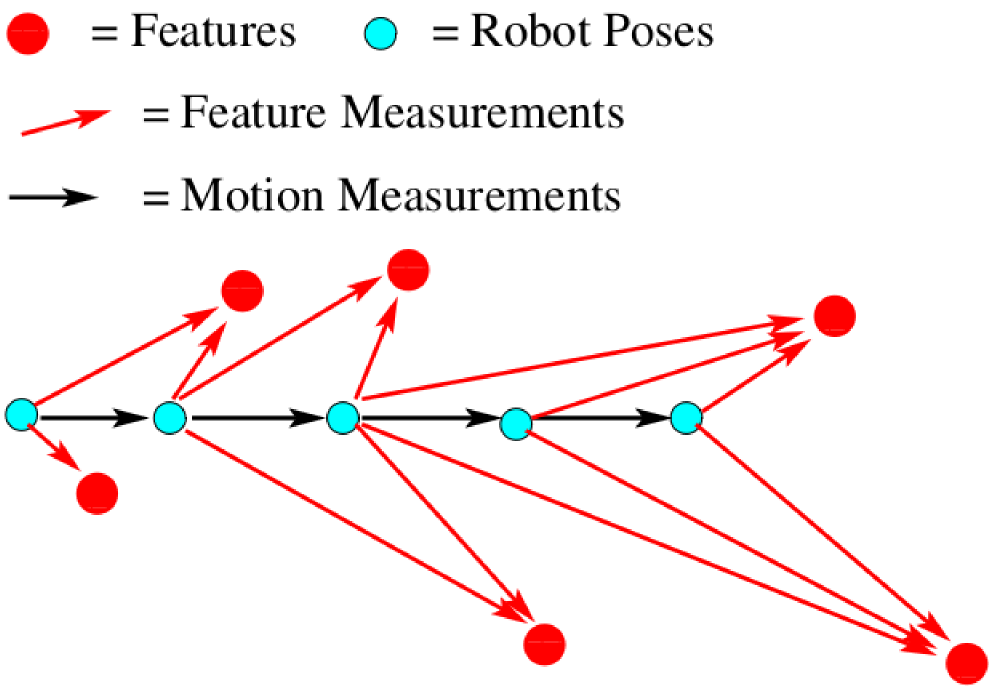
Measurement Matrix

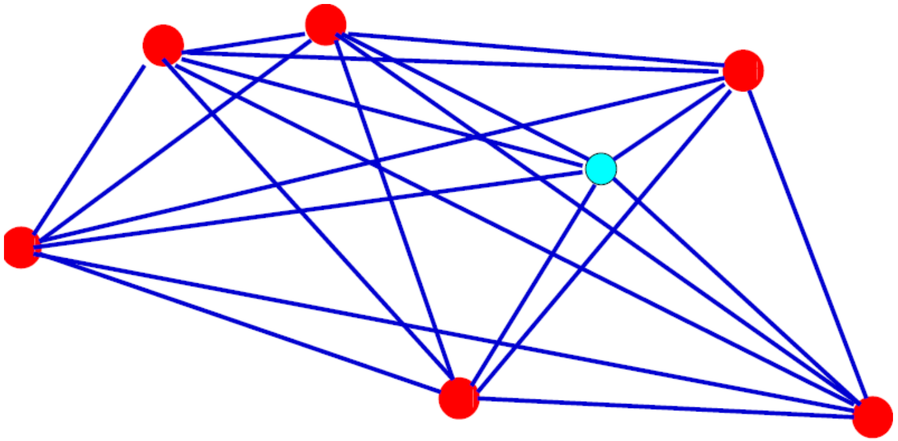


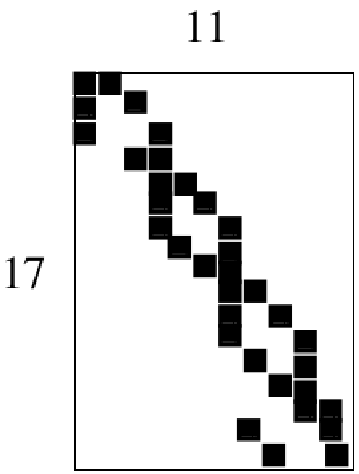
Information Matrix

Arrows do not indicate Bayes Net
Think each arrow has a factor node

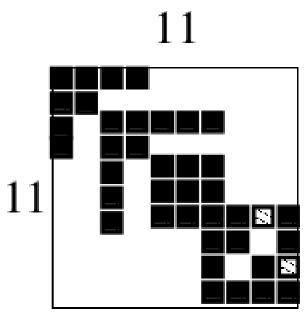
Sparsity



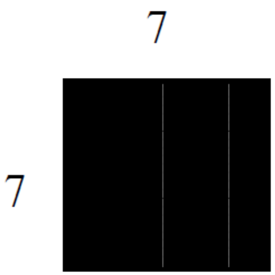




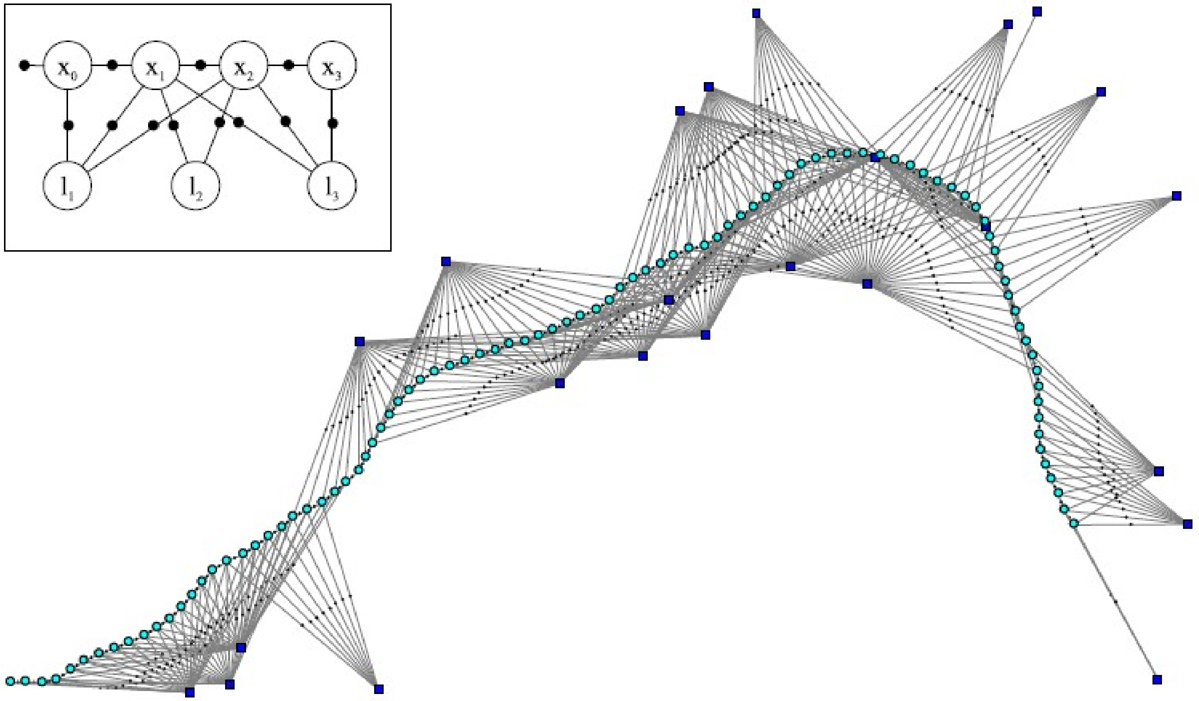
Measurement Matrix



Information Matrix



Covariance Matrix



- Eliminating all poses creates a fully connected feature graph.
- Eliminating all features creates a connected mess of pose nodes.
- If there are no loops working back and forth between features and poses along the path works.
- Loops require divide and conquer, local maps, star nodes, composite measurements...

QR Decomposition

Cholesky QR:

$A=QR$;

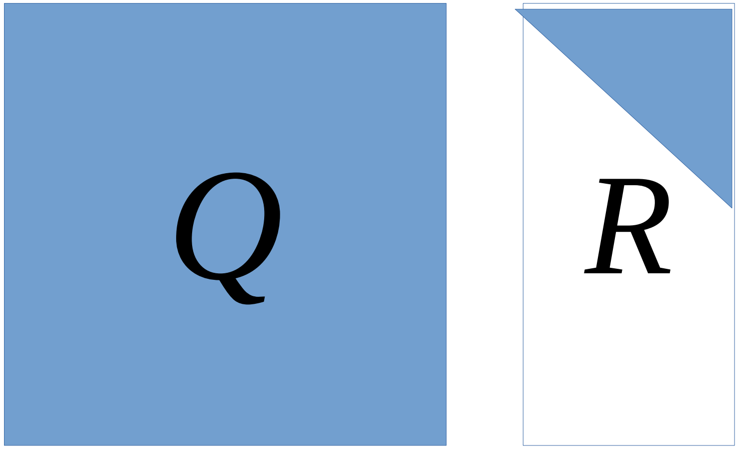
Q orthogonal ($Q^{T}Q=I$);

R upper triangular.

$$l(\mathbf{x}, \mathbf{c})= \tfrac{1}{2} \sum_{cij} ||A\boldsymbol{\Delta}\mathbf{x}\text{-}\mathbf{b}||^{2}$$

$$l(\mathbf{x}, \mathbf{c})= \tfrac{1}{2} \sum_{cij} ||Q^{T}A\boldsymbol{\Delta}\mathbf{x}\text{-}Q^{T}\mathbf{b}||^{2}$$

$$l(\mathbf{x}, \mathbf{c})= \tfrac{1}{2} \sum_{cij} ||R\boldsymbol{\Delta}\mathbf{x}\text{-}\mathbf{d}||^{2} + ||\mathbf{e}||^{2}$$



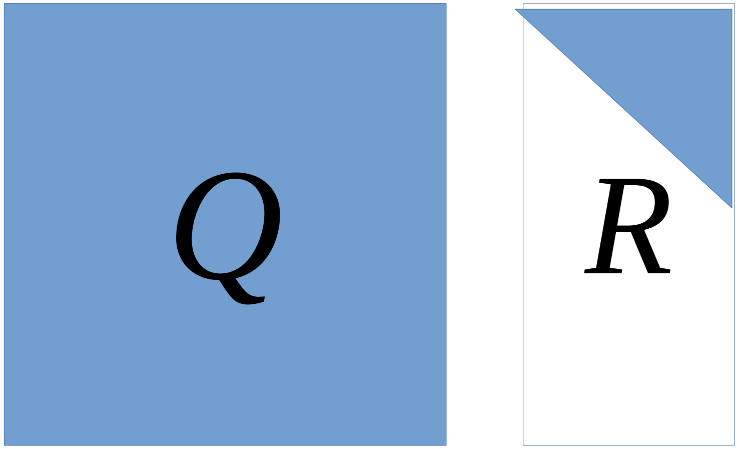
QR Decomposition

Cholesky QR:

$A=QR$;

Q orthogonal ($Q^TQ=I$);

R upper triangular.



$l(\mathbf{x}, \mathbf{c}) = \frac{1}{2} \sum_{cij} ||A\boldsymbol{\Delta}\mathbf{x}-\mathbf{b}||^2$

$l(\mathbf{x}, \mathbf{c}) = \frac{1}{2} \sum_{cij} ||Q^TA\boldsymbol{\Delta}\mathbf{x}-Q^T\mathbf{b}||^2$

$l(\mathbf{x}, \mathbf{c}) = \frac{1}{2} \sum_{cij} ||R\boldsymbol{\Delta}\mathbf{x}-\mathbf{d}||^2 + ||\mathbf{e}||^2$

$R\boldsymbol{\Delta}\mathbf{x} = \mathbf{d}$ is solved to make first term 0.

Last term tells us how good $\mathbf{c}$ was. (the data association)

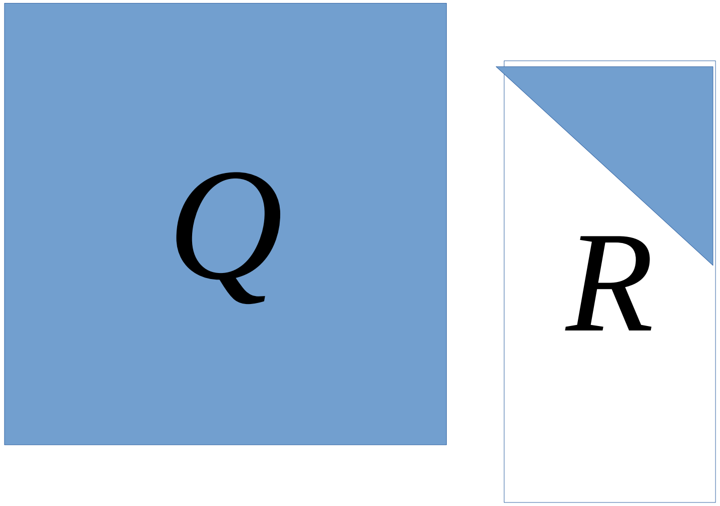
QR Decomposition

Cholesky QR:

$A=QR$;

Q orthogonal ($Q^TQ=I$);

R upper triangular.



$l(\mathbf{x}, \mathbf{c}) = \frac{1}{2} \sum_{cij} ||A\boldsymbol{\Delta}\mathbf{x}-\mathbf{b}||^2$

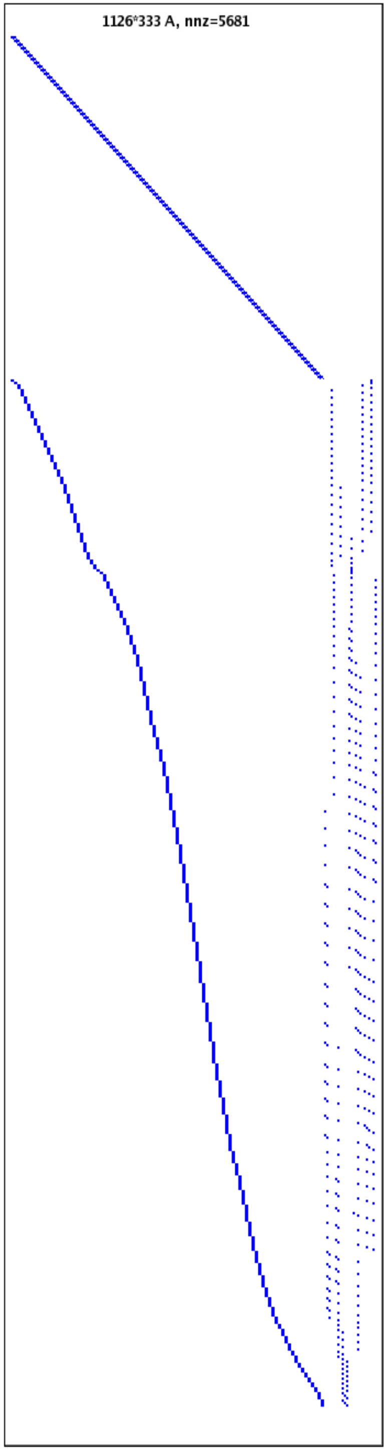
$l(\mathbf{x}, \mathbf{c}) = \frac{1}{2} \sum_{cij} ||Q^TA\boldsymbol{\Delta}\mathbf{x}-Q^T\mathbf{b}||^2$

$l(\mathbf{x}, \mathbf{c}) = \frac{1}{2} \sum_{cij} ||R\boldsymbol{\Delta}\mathbf{x}-\mathbf{d}||^2 + ||\mathbf{e}||^2$

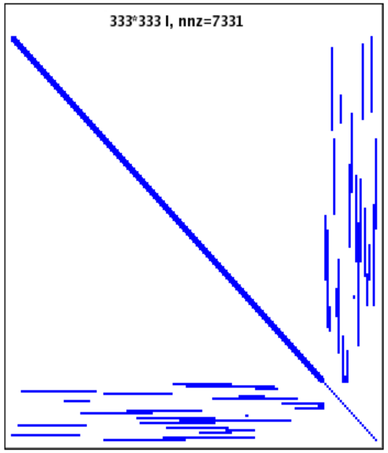
$R\boldsymbol{\Delta}\mathbf{x} = \mathbf{d}$ is solved to make first term 0.

Last term tells us how good $\mathbf{c}$ was. (the data association)

$(R^TR)^{-1} = R^{-1}R^{-T}$ is the Covariance Matrix (Marginalize by just ignoring rows/cols)

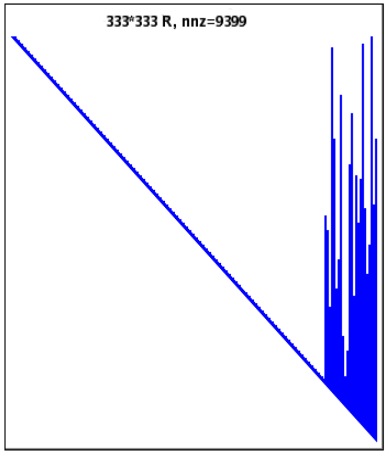


Measurement A



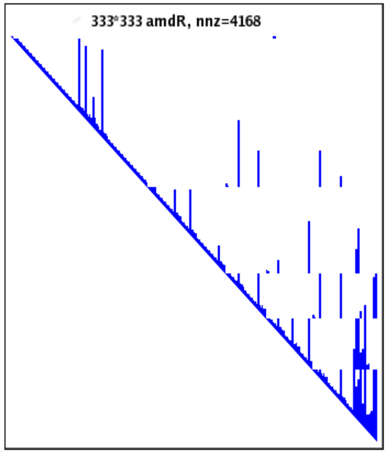
Top: Information Matrix

$$\Omega = A^T A$$



Middle: Cholesky R

$$A = QR;$$

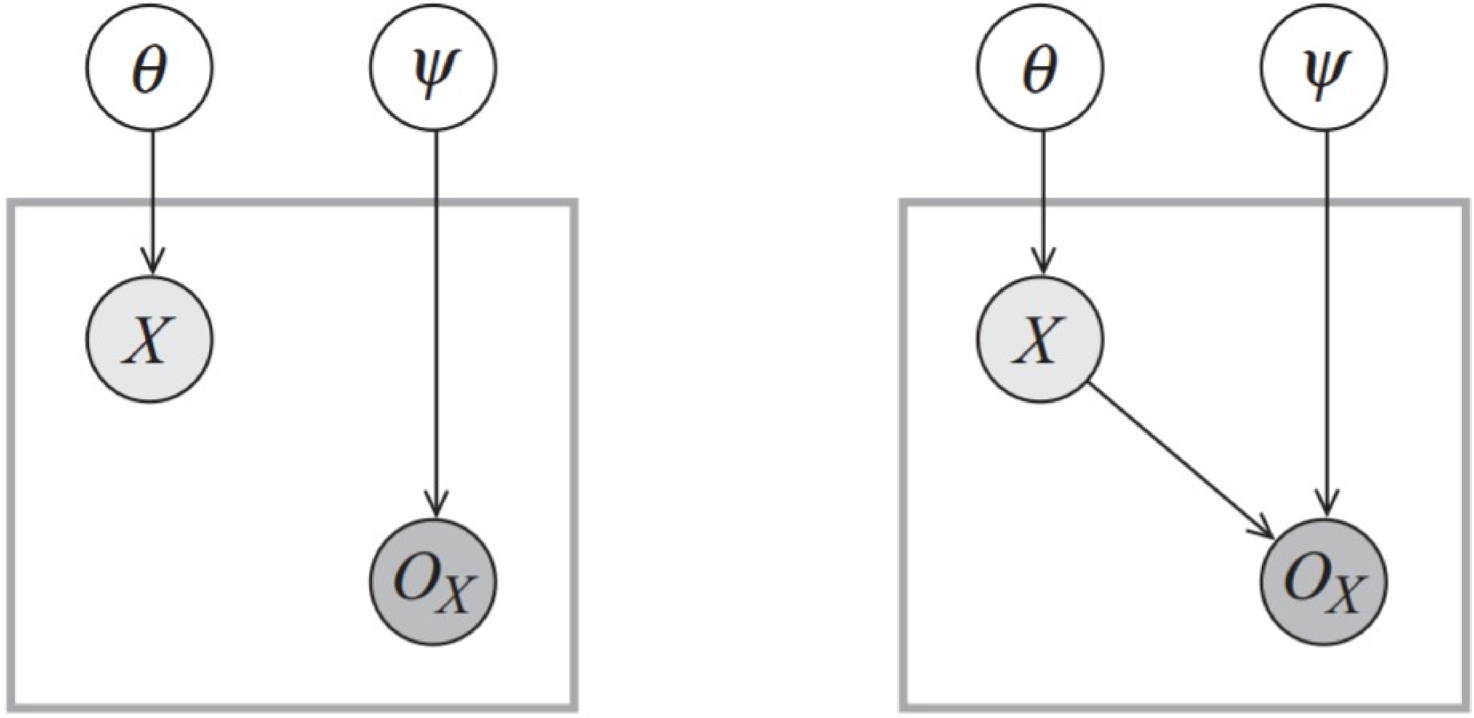


Bottom: R with better ordering (COLAMD)

From Dellaert and Kaess

Partially Observed Data (missing values)

- We could just marginalize over what we do not observe. (Allows throwing away incomplete samples)
- That works fine if the observation process is independent of the data. X is latent (hidden) here but we can see it if Ox is true. (lets just make X binary)



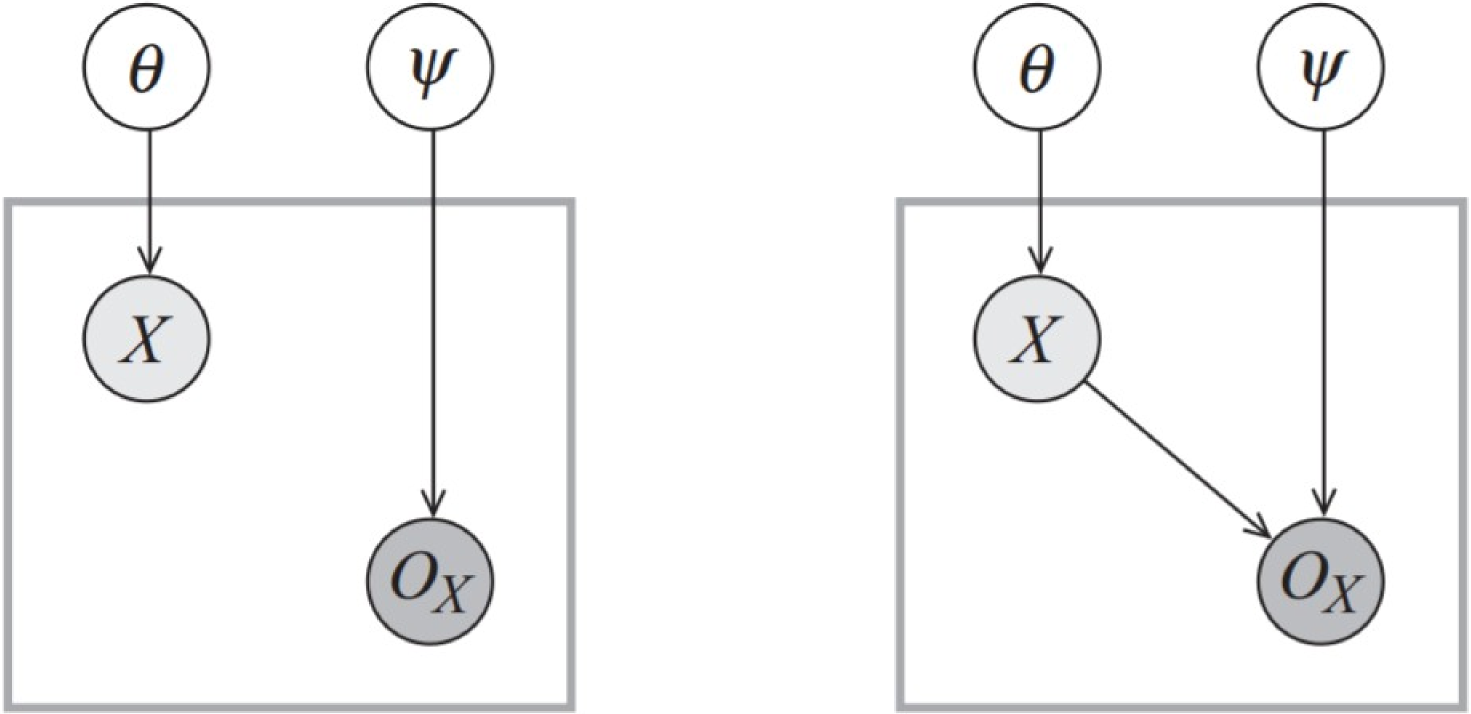
(a) Random missing values

(b) Deliberate missing values

Partially Observed Data (missing values)

- L(θ: D) = $\theta^{N_h}(1\text{-}\theta)^{N_t}\psi^{N_h+N_t}(1\text{-}\psi)^{N_?}$ vs.
- L(θ: D)=$(\theta\psi_h)^{N_{h(}}((1\text{-}\theta)\psi_t))^{N_t}[(\theta(1\text{-}\psi_h)+(1\text{-}\theta)(1\text{-}\psi_t)]^{N_?}$

Think of doing MLE now!

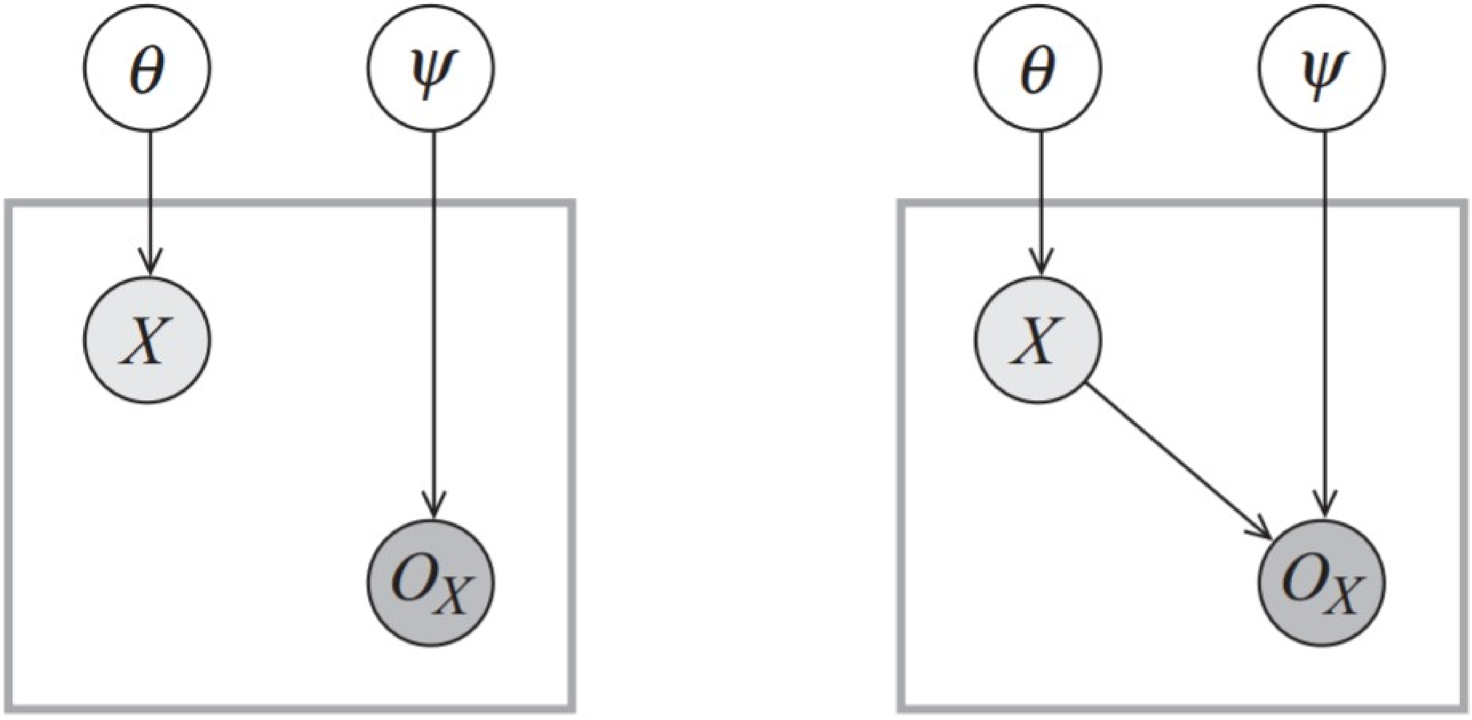


(a) Random missing values

(b) Deliberate missing values

Partially Observed Data (missing values)

- $\ln(L(\theta: D))=N_h\ln(\theta) + N_t\ln(1-\theta) + (N_h+N_t)\ln(\psi) + N_?\ln(1-\psi)$ vs.
- $\ln(L(\theta: D))=N_h\ln(\theta) + N_t\ln(1-\theta) + (N_h+N_t)\ln(\psi) + N_?\ln((\theta(1-\psi_h)+(1-\theta)(1-\psi_t)))$



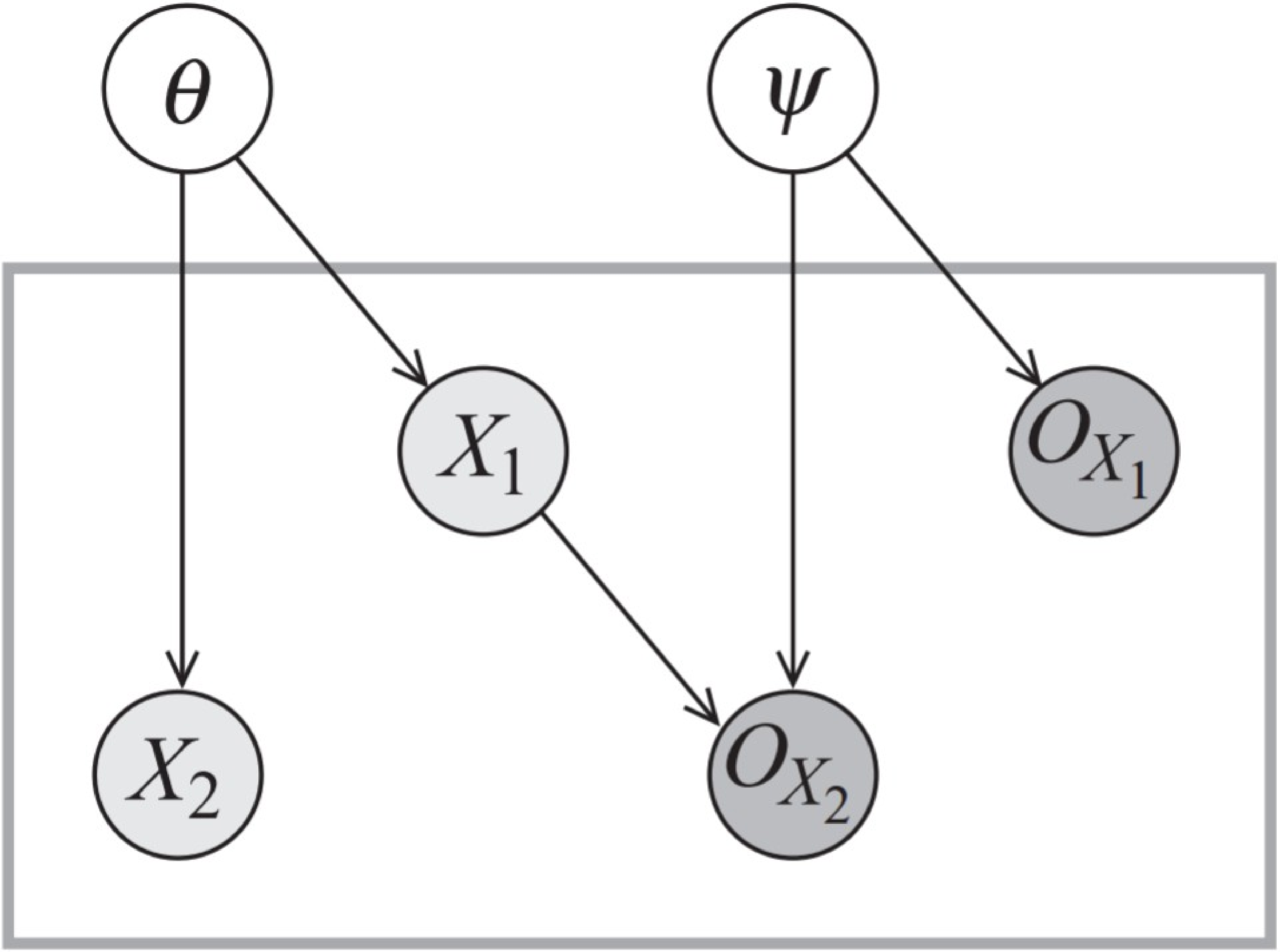
(a) Random missing values (b) Deliberate missing values

Decouple the Observation Process

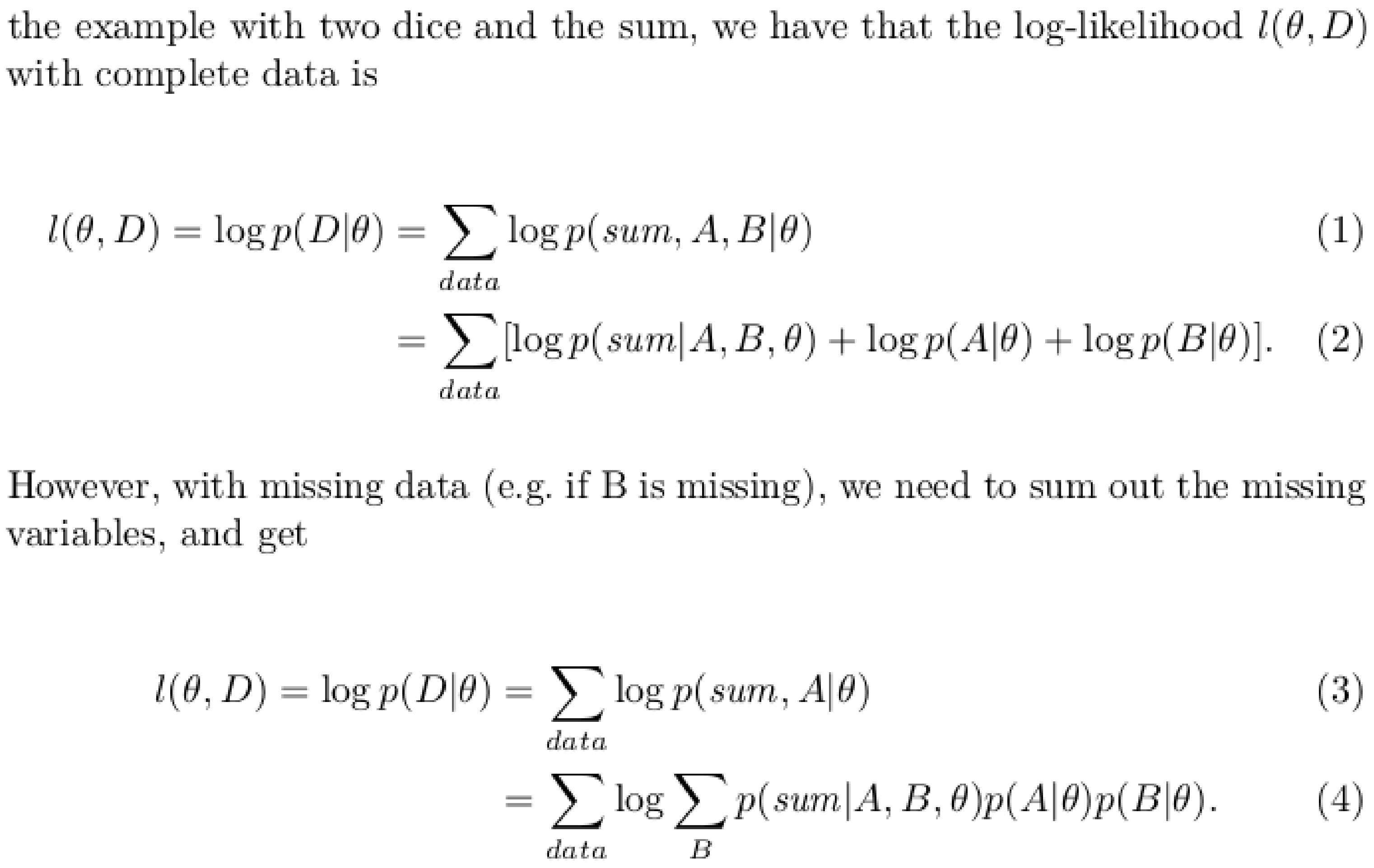
- MCAR: Missing completely at random, $O_x \perp \boldsymbol{X}$
- This is sufficient for decomposition of likelihood.
- Not necessary.
- MAR: Missing at random, $\boldsymbol{O}_x \perp \boldsymbol{X}_{hidden} \mid \boldsymbol{X}_{obs}$
- MAR allows us to factor the likelihood function:

  $L(\theta, \psi : D) = L(\theta : D)\, L(\psi : D)$
- MAR example: I do a series of experiments and randomly observe up to the first observation at which point I throw out the rest of the results even if there are more observations.

MAR: $\boldsymbol{O}_{x} \perp \boldsymbol{X}_{2} \mid \boldsymbol{X}_{1}$



Tutorial 5 - Partial Data



the example with two dice and the sum, we have that the log-likelihood $l(\theta, D)$ with complete data is

$$
\begin{aligned}
l(\theta, D) = \log p(D|\theta) &= \sum_{data} \log p(sum, A, B|\theta) && (1)\\
&= \sum_{data} [\log p(sum|A, B, \theta) + \log p(A|\theta) + \log p(B|\theta)]. && (2)
\end{aligned}
$$

However, with missing data (e.g. if B is missing), we need to sum out the missing variables, and get

$$
\begin{aligned}
l(\theta, D) = \log p(D|\theta) &= \sum_{data} \log p(sum, A|\theta) && (3)\\
&= \sum_{data} \log \sum_{B} p(sum|A, B, \theta)p(A|\theta)p(B|\theta). && (4)
\end{aligned}
$$

Suddenly just missing some data makes this hard or intractable for large data.
You will use EM to solve this sort of problem in the tutorial.
An alternative is Gradient Ascent.

MLE (partial observations)

- $L(\theta : D) = P(D \mid \theta) = \Pi_{\mathrm{m}} P(o[m] \mid \theta)$

  $= \Pi_{\mathrm{m}} \Sigma_{\text{parts of x not given in o[m]}} P(\boldsymbol{x}^m \mid \theta)$

  (ie set o[m] and sum out the rest of $\boldsymbol{x}$)

  Here a component of $\boldsymbol{x}^m$, say $x^m{}_i$, is either just a value if in $o[m]$ in D, or it is a variable that is summed over.

- This is a messy non-linear optimization, particularly since the o[m] can have different sets of observed variables.
- Key to gradient methods, on Bayes Nets, is that derivative of the factorized joint is ‘like dividing’ by the table values.

MLE (partial observations)

- $L(\theta : D) = P(D \mid \theta) = \Pi_m P(o[m] \mid \theta)$

  $= \Pi_m \Sigma_{\text{parts of x not given in o[m]}} P(\boldsymbol{x} \mid \theta)$

  (ie set o[m] and sum out the rest of $\boldsymbol{x}$)
- Book goes through how to take derivatives wrt table entries for CPD $P(\mathrm{x} \mid \boldsymbol{u})$ and then do gradient ascent.
- Trick is that any particular $\mathrm{P}(\boldsymbol{x} \mid \mathbf{u})$ appears in each term of the product with a sum of stuff multiplying to the right and to the left. By computing that ‘stuff’ smart one can save computations.
- Ends up using the clique tree inference with evidence done for each data.
- Nice is that the table entries all involve variables in some clique, (due to how we constructed the cliques).
- This is an application of that belief propagation with evidence.

Gradient Ascent

- There is a problem with maintaining a normalized probability.
  - Project all gradients on a hyperplane that satisfies the constraint. Then also enforce non negative

Or

  - Re-parametrize:

$$P(x \mid \boldsymbol{u}) = \exp(\ \lambda_{x|\mathbf{u}}\ ) \ /\ \Sigma_{x'} \exp(\ \lambda_{x'|\mathbf{u}}\ )$$

Or

  - Lagrange multipliers

Expectation Maximization EM

- For sanity I switch this to be less general:
- $\boldsymbol{z}$ are hidden (latent) variables (we called them $\boldsymbol{x}$ not part of $\mathrm{o}[m]$ before);
- $\boldsymbol{x}$ are observed (we called them $\mathrm{o}[m]$ before).

Expectation Maximization EM

- For sanity I switch this to be less general:
- $\boldsymbol{z}$ are hidden (latent) variables (we called them $\boldsymbol{x}$ not part of $\mathrm{o}[m]$ before);
- $\boldsymbol{x}$ are observed (we called them $\mathrm{o}[m]$ before).
- We define a sequence of parameter estimates $\boldsymbol{\theta}^t$

Expectation Maximization EM

- $\boldsymbol{z}$ are hidden (latent) variables (we called them $\boldsymbol{x}$ not part of $\mathrm{o}[m]$ before);
- $\boldsymbol{x}$ are observed (we called them $\mathrm{o}[m]$ before).
- E-Step: Compute the **E**xpected log-Likelihood of the data using current parameter estimate $\boldsymbol{\theta}^t$.

$$-\Sigma_{\mathbf{z}} \Sigma_{x \epsilon D}\, p(\mathbf{z}\, |\mathbf{x}, \boldsymbol{\theta}^t) \log p(\, \mathbf{x}, \mathbf{z} \mid \boldsymbol{\theta})$$

Expectation Maximization EM

- $\boldsymbol{z}$ are hidden (latent) variables (we called them $\boldsymbol{x}$ not part of $\mathrm{o}[m]$ before);
- $\boldsymbol{x}$ are observed (we called them $\mathrm{o}[m]$ before).
- E-Step: Compute the **E**xpected log-Likelihood of the data using current parameter estimate $\boldsymbol{\theta}^t$.

$$-\Sigma_{\mathbf{z}} \Sigma_{x \epsilon D}\, p(\mathbf{z}\, |\mathbf{x}, \boldsymbol{\theta}^t) \log p(\,\mathbf{x}, \mathbf{z} \mid \boldsymbol{\theta})$$

- M-Step Compute new parameters, $\boldsymbol{\theta}^{t+1}$, that **M**aximize that expectation wrt $\boldsymbol{\theta}$.

Loss

- We may want to learn parameters that minimize an expected loss.

  $E_{\theta}[loss(\boldsymbol{x}, \theta)] \approx (1/M)\sum_m \; loss(\boldsymbol{x}_m,\theta)$

- Max likelihood is $loss(\boldsymbol{x}, \theta) = -\log p_{\theta}(\boldsymbol{x})$
- Conditional loss is $loss(\boldsymbol{x}, \boldsymbol{y}, \theta) = -\log p_{\theta}(\boldsymbol{x} | \boldsymbol{y})$

  (x could be segmentation variable)

- Another loss might be classification error or risk of an particular miss classification.

EM vs Gradient Ascent

- EM tends to take a big first step.
- Gradient ascent tends to move slowly at first.
- EM
  - is for this exact problem;
  - will never move to a worse or invalid estimate.
- Gradient ascent is a generic method that has many standard implementations using various 'tricks'.
- Both get stuck in local max.

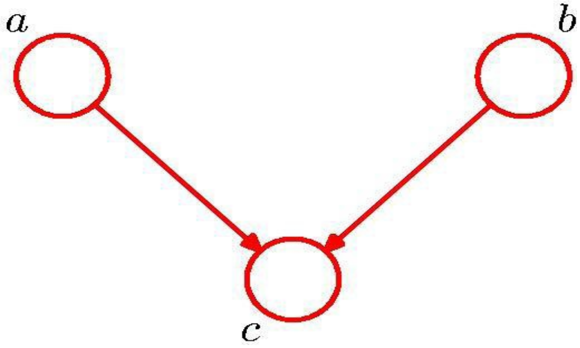
Bias and Variance

- Our choice of possible distributions can cause a bias.
- Too small an 'hypothesis space' and even infinite data might not give a perfect distribution match.
- Too general a model can lead to over fitting and variance with small amount of data (ie next data point changes distribution a lot.)
- Ideal is if we can constraint the distribution space in ways that we expect to be valid.

Identifiable

- $\theta \neq \theta' \Rightarrow P(\mathbf{x} \mid \theta) \neq P(\mathbf{x} \mid \theta')$
- This contrasts with sufficient stats which was different data could give same parameters θ.
- Here different parameters could both end up giving the exact same distribution for our observed variables.
- Example of not identifiable: 2 correlated coin flips (a 1 and 5 kr coin) where we only get to see one at random. We can learn the marginal P of each but not the correlation. Thus we could for example estimate:
  - **P(1 head)** = ratio of 1 heads to total observed 1
  - **P(5 head | 1 head) P(1 head)** + **P(5 head | 1 tail) (1-P(1 head)) =** ratio of 5 heads to total observed 5
  - **(1-P(5 head | 1 head)) P(1 head)** + (1-**P(5 head | 1 tail)) (1-P(1 head)) =** ratio of 5 tails to total observed 5
- Thus 3 equations and 3 unknowns seems ok but the last two are linearly dependent (actually the same equation).
- Also can be problems when the model allows arbitrary choices for hidden values giving different parameter values (renaming) but same observation likelihood. Imagine knowing sets of all the measurements from each coin but not knowing the labels for the sets (ie 1 or 5 kr)
- There is a concept of local identifiable if the parameters are in a local max in the parameter space.

Tutorial 5 – EM - Partial Data



- Here a b and c are binary so 12 table values to learn (including the priors on a and b).
- Critical concept is again ‘Sufficient Statistics’
- In the E step we need to compute the expected likelihood given our current tables.
- This is done by means of the expected sufficient stats where we compute expectations over our missing data.
- What were the sufficient stats for the binomial dist?
- We then get an **E**xpected likelihood in terms of our parameters which we can **M**aximize...

Latent Variables

- Latent variables are hidden variables
- They might choose/index over another distribution.
- Gaussian Mixture models are popular.

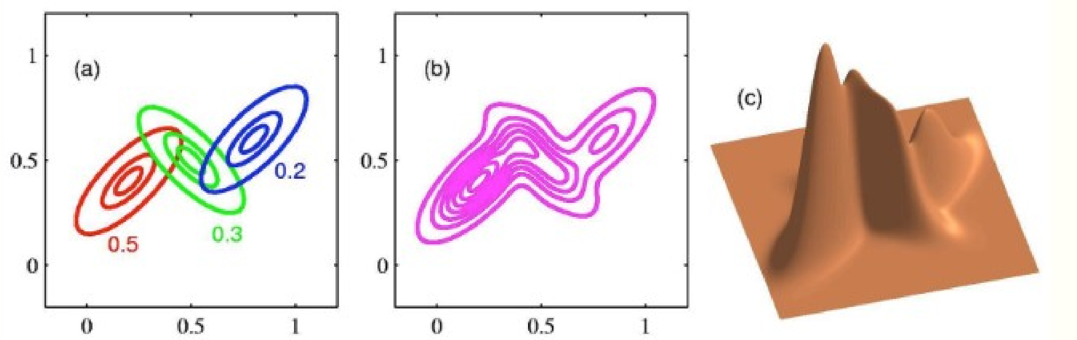
$p(\boldsymbol{x}, z) = p(\boldsymbol{x} \mid z)P(z)$

z is an index 1, 2, 3, . . ., K

$\pi_k \equiv P(z=k)$

$p(\boldsymbol{x} \mid z)$ is a Gaussian over $\boldsymbol{x}$ with mean and variance that depends on z.

$$p(x) = \sum^{K} p(x|z=k)p(z=k) = \sum^{K} \pi_k \mathcal{N}(x; \mu_k, \Sigma_k).$$

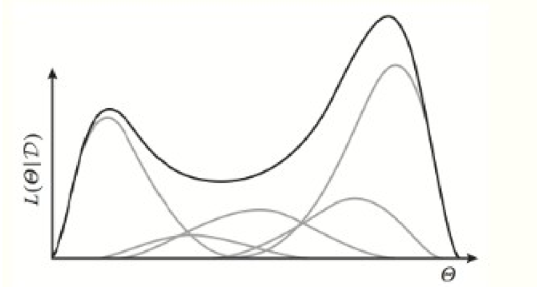


Gaussian Mixture Learning

- We want to maximize our data likelihood:

$$\log p(D) = \sum_{x \in D} \log p(x) = \sum_{x \in D} \log\left(\sum_{z} p(x|z)p(z) \right)$$

- This will not factor nor have a single mode:(



EM GM Learning

- E-Step: compute the probability of the latent variable:

$$a^t_{xk} = p(z_k \mid \boldsymbol{x};\ \pi^t_k\ \boldsymbol{\mu}^t_k,\ \Sigma^t_k) \propto N(\boldsymbol{x};\ \boldsymbol{\mu}^t_k,\ \Sigma^t_k) p(z_k;\ \pi^t_k\ \boldsymbol{\mu}^t_k,\ \Sigma^t_k)$$

Recall from previous slide:

- E-Step: Compute the **E**xpected log-Likelihood of the data using current parameter estimate $\boldsymbol{\theta}^t$.

$$-\sum_{\boldsymbol{z}} \sum_{x \epsilon D} p(\boldsymbol{z} \mid \boldsymbol{x},\ \boldsymbol{\theta}^t) \log p(\ \boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\theta})$$

EM GM Learning

- E-Step: compute the probability of the latent variable:

$$a^t_{xk} = p(z_k \mid \boldsymbol{x};\, \pi^t_k\, \boldsymbol{\mu}^t_k\,,\, \Sigma^t_k) \propto N(\boldsymbol{x};\, \boldsymbol{\mu}^t_k,\, \Sigma^t_k) p(z_k;\, \pi^t_k\, \boldsymbol{\mu}^t_k\,,\, \Sigma^t_k)$$

- the expected loglikelihood of the data given a^t_{xk}.

$$= \Sigma_{\mathrm{k}} \Sigma_{x \epsilon D}\, a^t_{xk} \,\{\log p(\, \boldsymbol{x} \mid z_k\,;\, \pi_k\, \boldsymbol{\mu}_k\,,\, \Sigma_{\mathrm{k}}) + \log p(z_k\,;\, \pi_k,\, \boldsymbol{\mu}_k\,,\, \Sigma_k)\}$$

Interpret ; as ‘using’
or ‘with’.

EM GM Learning

- E-Step: compute the probability of the latent variable:

$$a^t_{xk} = p(z_k \mid \boldsymbol{x}; \pi^t_k\, \boldsymbol{\mu}^t_k, \Sigma^t_k) \propto N(\boldsymbol{x}; \boldsymbol{\mu}^t_k, \Sigma^t_k) p(z_k; \pi^t_k\, \boldsymbol{\mu}^t_k, \Sigma^t_k)$$

- the expected loglikelihood of the data given a^t_{xk}.

$$= \Sigma_{\mathrm{k}} \Sigma_{x \epsilon D}\, a^t_{xk} \{\log p(\boldsymbol{x} \mid z_k; \pi_k\, \boldsymbol{\mu}_k, \Sigma_{\mathrm{k}}) + \log p(z_k; \pi_k, \boldsymbol{\mu}_k, \Sigma_k)\}$$

$$= \Sigma_{\mathrm{k}} \Sigma_{x \epsilon D}\, a^t_{xk} \{\log N(\boldsymbol{x}; \boldsymbol{\mu}_{\mathrm{k}}, \Sigma_{\mathrm{k}}) + \log \pi_{\mathrm{k}}\}$$

EM GM Learning

- E-Step: compute the probability of the latent variable:

$$a^t_{xk} = p(z_k \mid \boldsymbol{x};\ \pi^t_k\ \boldsymbol{\mu}^t_k,\ \Sigma^t_k) \propto N(\boldsymbol{x};\ \boldsymbol{\mu}^t_k,\ \Sigma^t_k)\, p(z_k;\ \pi^t_k\ \boldsymbol{\mu}^t_k,\ \Sigma^t_k)$$

- the expected loglikelihood of the data given a^t_{xk}.

$$= \Sigma_k \Sigma_{x \epsilon D}\ a^t_{xk}\ \{\log p(\ \boldsymbol{x}\ |z_k;\ \pi_k\ \boldsymbol{\mu}_k,\ \Sigma_k) + \log p(z_k;\ \pi_k,\ \boldsymbol{\mu}_k,\ \Sigma_k)\}$$

$$= \Sigma_k \Sigma_{x \epsilon D}\ a^t_{xk}\ \{\log N(\ \boldsymbol{x};\ \boldsymbol{\mu}_k,\ \Sigma_k) + \log \pi_k\}$$

$$= \Sigma_k \Sigma_{x \epsilon D}\ a^t_{xk}\ \{(-½)(\boldsymbol{x}\text{-}\ \boldsymbol{\mu}_k)^T \Sigma_k^{-1} (\boldsymbol{x}\text{-}\ \boldsymbol{\mu}_k)\ \ -\ (½)\ \log |2\pi\Sigma_k| + \log \pi_k\}$$

M-Step: Max with

EM GM Learning

- E-Step: compute the probability of the latent variable:

$$a^t_{xk} = p(z_k \mid \boldsymbol{x}; \pi^t_k\, \boldsymbol{\mu}^t_k, \Sigma^t_k) \propto N(\boldsymbol{x}; \boldsymbol{\mu}^t_k, \Sigma^t_k)\, p(z_k; \pi^t_k\, \boldsymbol{\mu}^t_k, \Sigma^t_k)$$

- the expected loglikelihood of the data given a^t_{xk}.

$$= \Sigma_k \Sigma_{x \epsilon D}\, a^t_{xk} \{\log p(\boldsymbol{x} \mid z_k; \pi_k\, \boldsymbol{\mu}_k, \Sigma_k) + \log p(z_k; \pi_k, \boldsymbol{\mu}_k, \Sigma_k)\}$$

$$= \Sigma_k \Sigma_{x \epsilon D}\, a^t_{xk} \{\log N(\boldsymbol{x}; \boldsymbol{\mu}_k, \Sigma_k) + \log \pi_k\}$$

$$= \Sigma_k \Sigma_{x \epsilon D}\, a^t_{xk} \{(-\tfrac{1}{2})(\boldsymbol{x} - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_k) - (\tfrac{1}{2}) \log |2\pi\Sigma_k| + \log \pi_k\}$$

M-Step: Max it!

let $c_k = \Sigma_{x \epsilon D}\, a^t_{xk}$; $\max_{\pi} (\Sigma_k c_k \log \pi_k)$ subject to $\Sigma_k \pi_k = 1$.

EM GM Learning

- E-Step: compute the probability of the latent variable:

$$a^t_{xk} = p(z_k \mid \boldsymbol{x}; \pi^t_k\, \boldsymbol{\mu}^t_k, \Sigma^t_k) \propto N(\boldsymbol{x}; \boldsymbol{\mu}^t_k, \Sigma^t_k) p(z_k; \pi^t_k\, \boldsymbol{\mu}^t_k, \Sigma^t_k)$$

- the expected loglikelihood of the data given a^t_{xk}.

$$= \Sigma_{\mathrm{k}} \Sigma_{x \epsilon D}\, a^t_{xk} \{\log p(\boldsymbol{x} \mid z_k; \pi_k\, \boldsymbol{\mu}_k, \Sigma_{\mathrm{k}}) + \log p(z_k; \pi_k, \boldsymbol{\mu}_k, \Sigma_k)\}$$

$$= \Sigma_{\mathrm{k}} \Sigma_{x \epsilon D}\, a^t_{xk} \{\log N(\boldsymbol{x}; \boldsymbol{\mu}_{\mathrm{k}}, \Sigma_{\mathrm{k}}) + \log \pi_{\mathrm{k}}\}$$

$$= \Sigma_{\mathrm{k}} \Sigma_{x \epsilon D}\, a^t_{xk} \{(-½)(\boldsymbol{x} - \boldsymbol{\mu}_{\mathrm{k}})^{\mathrm{T}} \Sigma_{\mathrm{k}}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_{\mathrm{k}}) - (½) \log |2\pi\Sigma_{\mathrm{k}}| + \log \pi_{\mathrm{k}}\}$$

let $c_k = \Sigma_{x \epsilon D}\, a^t_{xk}$; $\max_\pi (\Sigma_{\mathrm{k}} c_k \log \pi_{\mathrm{k}})$ subject to $\Sigma_{\mathrm{k}} \pi_{\mathrm{k}} = 1$.

$$\Sigma_{\mathrm{k}} c_k \log \pi_{\mathrm{k}} - \lambda(\Sigma_{\mathrm{k}} \pi_{\mathrm{k}} - 1) \Rightarrow c_k / \pi_{\mathrm{k}} - \lambda = 0; \quad \Sigma_{\mathrm{k}} \pi_{\mathrm{k}} - 1 = 0$$

Notice: We do not consider that a^t_{xk} depends on π^t_{k}.

EM GM Learning

- E-Step: compute the probability of the latent variable:

$$a^t_{xk} = p(z_k \mid \boldsymbol{x};\, \pi^t_k\, \boldsymbol{\mu}^t_k,\, \Sigma^t_k) \propto N(\boldsymbol{x};\, \boldsymbol{\mu}^t_k,\, \Sigma^t_k)\, p(z_k;\, \pi^t_k\, \boldsymbol{\mu}^t_k,\, \Sigma^t_k)$$

- the expected loglikelihood of the data given a^t_{xk}.

$$= \Sigma_k \Sigma_{x\epsilon D}\, a^t_{xk}\, \{\log p(\boldsymbol{x} \mid z_k;\, \pi_k\, \boldsymbol{\mu}_k,\, \Sigma_k) + \log p(z_k;\, \pi_k,\, \boldsymbol{\mu}_k,\, \Sigma_k)\}$$

$$= \Sigma_k \Sigma_{x\epsilon D}\, a^t_{xk}\, \{\log N(\boldsymbol{x};\, \boldsymbol{\mu}_k,\, \Sigma_k) + \log \pi_k\}$$

$$= \Sigma_k \Sigma_{x\epsilon D}\, a^t_{xk}\, \{(-½)(\boldsymbol{x} - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_k) - (½) \log |2\pi\Sigma_k| + \log \pi_k\}$$

let $c_k = \Sigma_{x\epsilon D}\, a^t_{xk}$; $\max_\pi (\Sigma_k c_k \log \pi_k)$ subject to $\Sigma_k \pi_k = 1$.

$$\Sigma_k c_k \log \pi_k - \lambda(\Sigma_k \pi_k - 1) \Rightarrow c_k / \pi_k - \lambda = 0; \quad \Sigma_k \pi_k - 1 = 0$$

$$\Rightarrow 1 = \Sigma_k \pi_k = \Sigma_k c_k / \lambda \Rightarrow \lambda = 1 / \Sigma_k c_k \Rightarrow \pi_k = c_k / \Sigma_k c_k$$

$\Sigma_k c_k = M$, if we have normalized the a’s.

EM GM Learning

- E-Step: compute the probability of the latent variable:

$$a^t_{xk} = p(z_k \mid \boldsymbol{x}; \pi^t_k\, \boldsymbol{\mu}^t_k, \Sigma^t_k) \propto N(\boldsymbol{x}; \boldsymbol{\mu}^t_k, \Sigma^t_k)\, p(z_k; \pi^t_k\, \boldsymbol{\mu}^t_k, \Sigma^t_k)$$

- the expected loglikelihood of the data given a^t_{xk}.

$$= \Sigma_k \Sigma_{x\epsilon D}\, a^t_{xk} \{\log p(\boldsymbol{x} \mid z_k; \pi_k\, \boldsymbol{\mu}_k, \Sigma_k) + \log p(z_k; \pi_k, \boldsymbol{\mu}_k, \Sigma_k)\}$$

$$= \Sigma_k \Sigma_{x\epsilon D}\, a^t_{xk} \{(-\tfrac{1}{2})(\boldsymbol{x}- \boldsymbol{\mu}_k)^T \Sigma_k^{-1} (\boldsymbol{x}- \boldsymbol{\mu}_k) - (\tfrac{1}{2}) \log |2\pi\Sigma_k| + \log \pi_k\}$$

- M- Step: Max it: $\pi^{t+1}_k = c_k/M;$ $c_k = \Sigma_{x'\epsilon D}\, a^t_{x'k};$

$$\boldsymbol{\mu}^{t+1}_k = \Sigma_{x\epsilon D}\, (a^t_{xk} / c_k)\, \boldsymbol{x};$$

$$\Sigma^{t+1}_k = \Sigma_{x\epsilon D}\, (a^t_{xk} / c_k)\, (\boldsymbol{x}-\boldsymbol{\mu}^{t+1}_k)(\boldsymbol{x}-\boldsymbol{\mu}^{t+1}_k)^T;$$

EM GM Learning

- E-Step: compute the probability of the latent variable:

$$a^t_{xk} = p(z_k \mid \boldsymbol{x}; \pi^t_k\, \boldsymbol{\mu}^t_k, \Sigma^t_k) \propto N(\boldsymbol{x}; \boldsymbol{\mu}^t_k, \Sigma^t_k) p(z_k; \pi^t_k\, \boldsymbol{\mu}^t_k, \Sigma^t_k)$$

- the expected loglikelihood of the data given a^t_{xk}.

$$= \Sigma_k \Sigma_{x\epsilon D}\; a^t_{xk} \{\log p(\boldsymbol{x} \mid z_k; \pi_k\, \boldsymbol{\mu}_k, \Sigma_k) + \log p(z_k; \pi_k, \boldsymbol{\mu}_k, \Sigma_k)\}$$

- M- Step: Max it: $\pi^{t+1}_k = c_k / \mathrm{M}$; $\boldsymbol{\mu}^{t+1}_k = \Sigma_{x\epsilon D} (a^t_{xk} / \boldsymbol{c}_k)\, \boldsymbol{x}$

$$\Sigma^{t+1}_k = \Sigma_{x\epsilon D} (a^t_{xk} / c_k) (\boldsymbol{x} - \boldsymbol{\mu}^{t+1}_k)(\boldsymbol{x} - \boldsymbol{\mu}^{t+1}_k)^T; \qquad c_k = \Sigma_{x\epsilon D}\, a^t_{xk};$$

- Chose $a^t_{xk} = 1$ *for* $z_k = \arg\max_{zk}\; p(z_k \mid \boldsymbol{x}; \pi^t_k\, \boldsymbol{\mu}^t_k, \Sigma^t_k)$; else $a^t_{xk} = 0$

  and you have k-means.