DS-GA 3001.008 Modelling time series data

L3. Latent state models 1: Kalman filtering

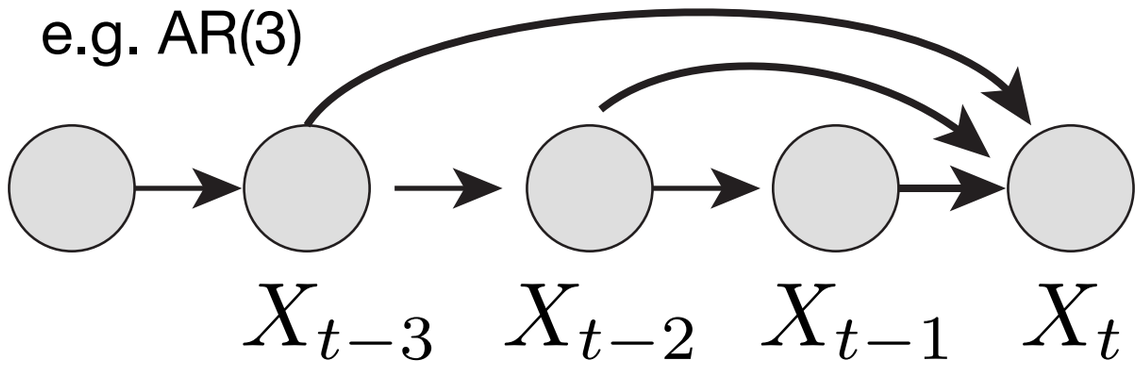
Instructor: Cristina Savin

NYU, CNS & CDS

So far : **directly** model the dependencies across time, under certain **restrictions.**

BUT only for **scalar** variables really

ARIMA



linear dependencies
stationary*

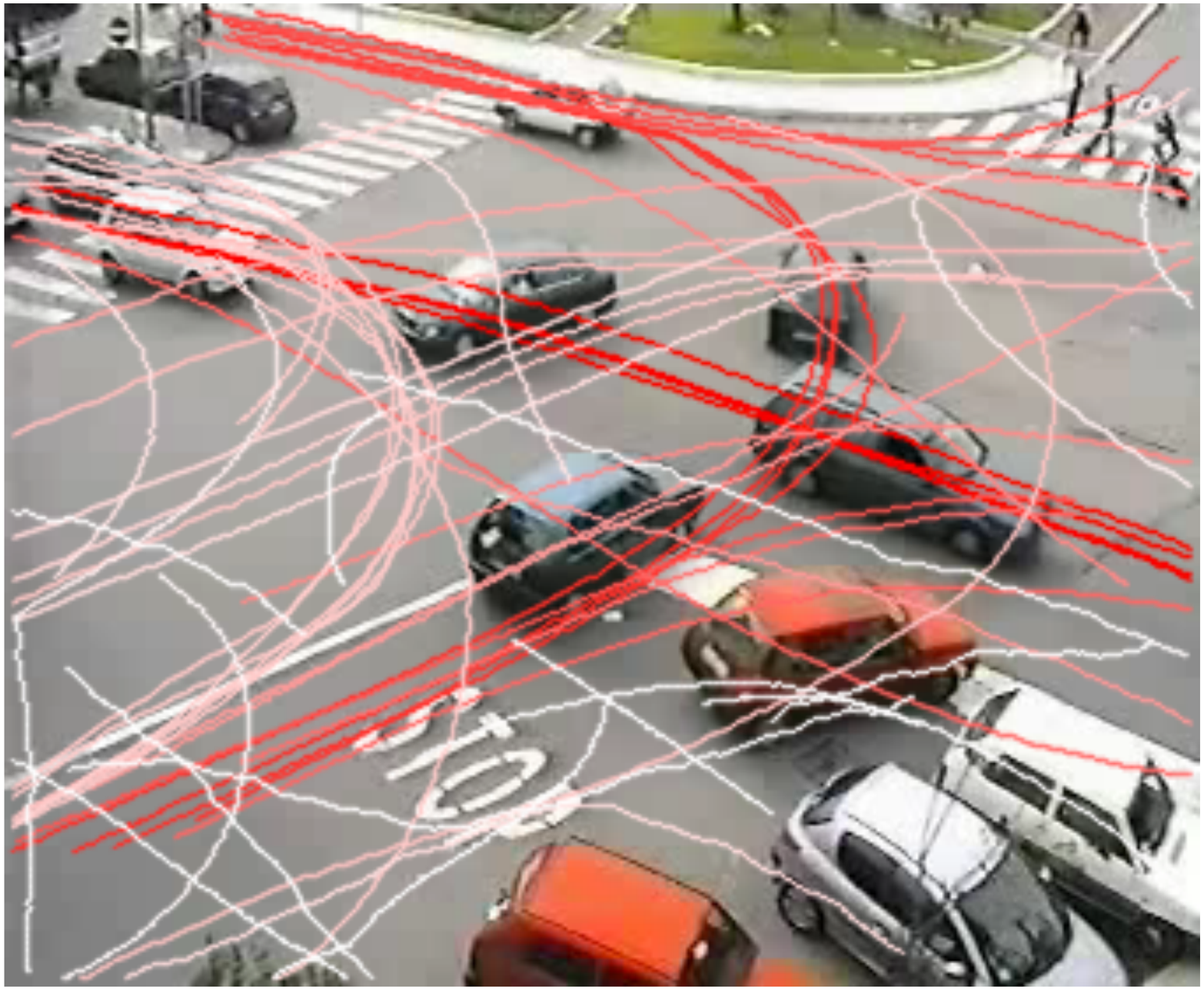
Tractable

The polynomial representation provides simple characterization of the properties of the corresponding stochastic process:

- The roots of the polynomial determine **causality** and **invertibility**, etc.
- They also provide a way to simplify complex models to **minimal form.**
- We can also use the roots to do **prediction**
- Polynomial form defines recursion expressions for the moments of the underlying distribution, which we can use to estimate model **parameters.**

*Polynomial dependence of the mean, for AR**I**MA

Most interesting data is complex and **multidimensional**



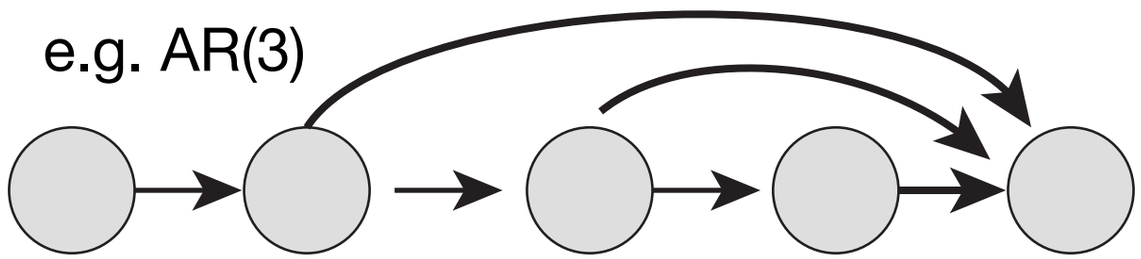
From Saunier & Sayed 2008

Directly model the dependencies across time, under certain **restrictions**.

Tractable

ARIMA

e.g. AR(3)

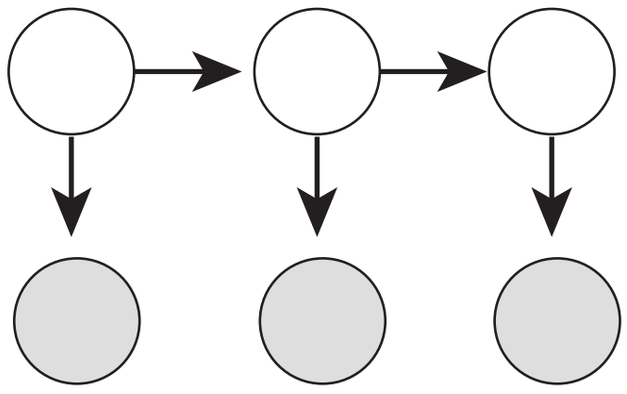


linear dependencies
stationary*
scalar variables

In the latent space the temporal dependencies are simple:

1st order Markov

Latent state models



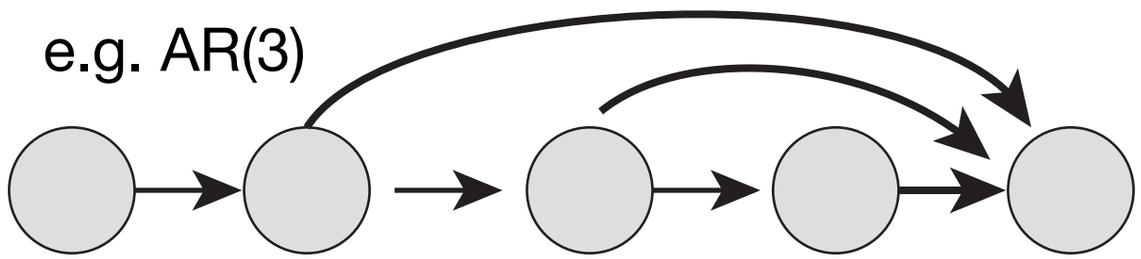
Goal: find a **latent** variable
that summarizes
the relevant **history**
while keeping math simple

Directly model the dependencies across time, under certain **restrictions**.

Tractable

ARIMA

e.g. AR(3)

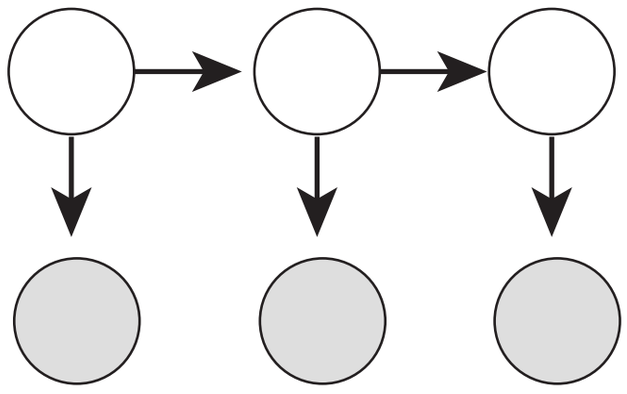


linear dependencies
stationary*
scalar variables

In the latent space the temporal dependencies are simple:

1st order Markov

Latent state models

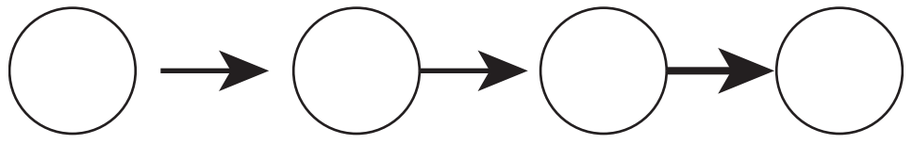


Goal: find a **latent** variable
that summarizes
the relevant **history**
while keeping math simple

The latent variable is either discrete for **hidden Markov models** (HMMs) or continuous for **latent dynamical systems** (LDS)

Markov property

e.g. first order Markov process



A sequence has the Markov property if, once we know the most recent $\mathbf{x}_{t-1}$ the current state $\mathbf{x}_t$ is independent on the past history.

$$\mathrm{P}\left(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_{t-2}, \cdots, \mathbf{x}_0\right) = \mathrm{P}(\mathbf{x}_t|\mathbf{x}_{t-1})$$

This means that we have a simple factorization for the full sequence:

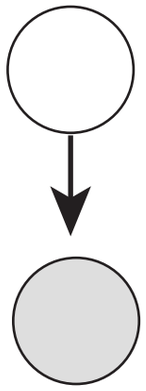
$$\mathrm{P}\left(\mathbf{x}_t, \mathbf{x}_{t-1}, \cdots, \mathbf{x}_0\right) = \prod_k \mathrm{P}\left(\mathbf{x}_k|\mathbf{x}_{k-1}\right)$$

Nice if individual components are simple (exponential family)

Reminder: chain rule

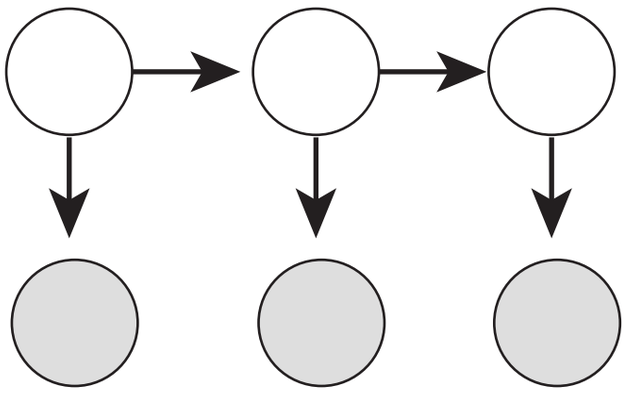
$$\mathrm{P}\left(\mathbf{x}_t, \mathbf{x}_{t-1}, \cdots, \mathbf{x}_0\right) = \prod_k \mathrm{P}\left(\mathbf{x}_k|\mathbf{x}_{k-1}, \cdots, \mathbf{x}_0\right)$$

Kalman filtering motivation



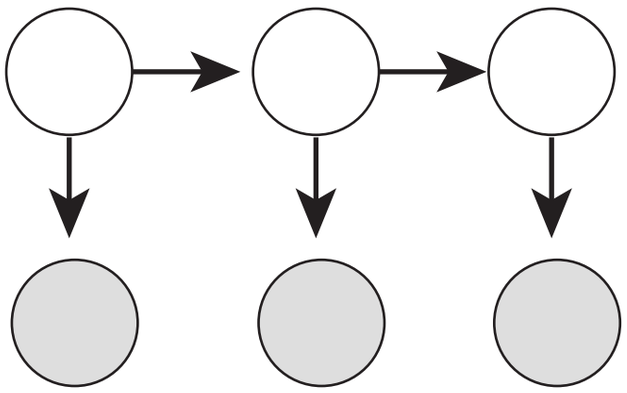
If I want to determine the value of a variable z,
based on a noisy measurement x,
I just take multiple measurements to get
a better estimate

Kalman filtering motivation



If I want to determine the value of a variable z,
based on a noisy measurement x,
I just take multiple measurements to get
a better estimate

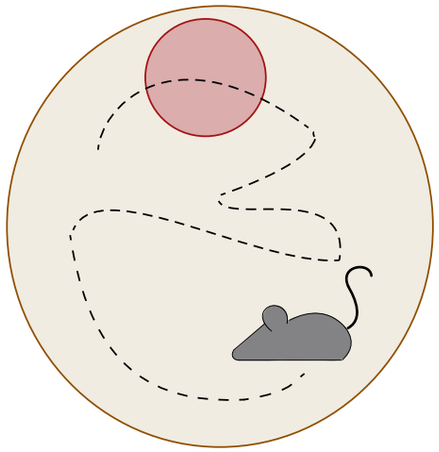
Kalman filtering motivation

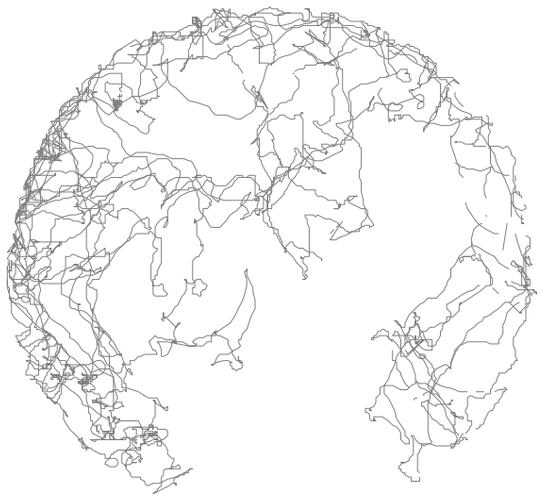


If I want to determine the value of a variable z,
based on a noisy measurement x,
I just take multiple measurements to get
a better estimate

Can’t do that of z changes over time,
but there is still hope if z changes slowly

Applications include: tracking objects (e.g., people, hands), navigation
Computer vision: tracking features in video, stabilizing depth measurements,
and many more

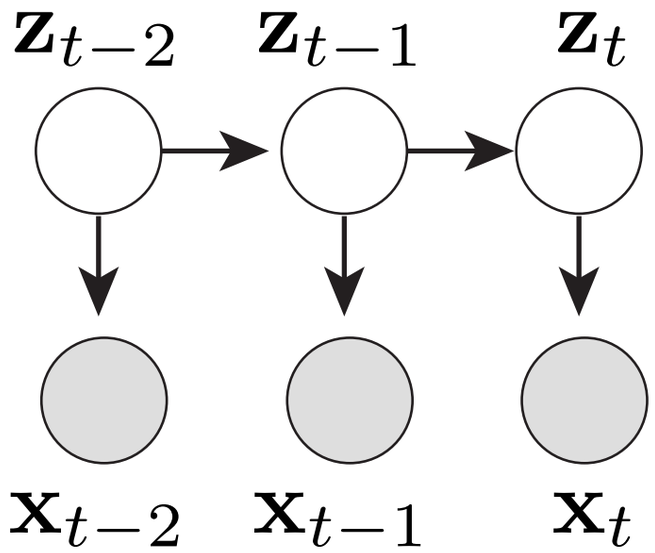




E.g. noisy measurements
of the animal's
position from top camera

OpenCV demo

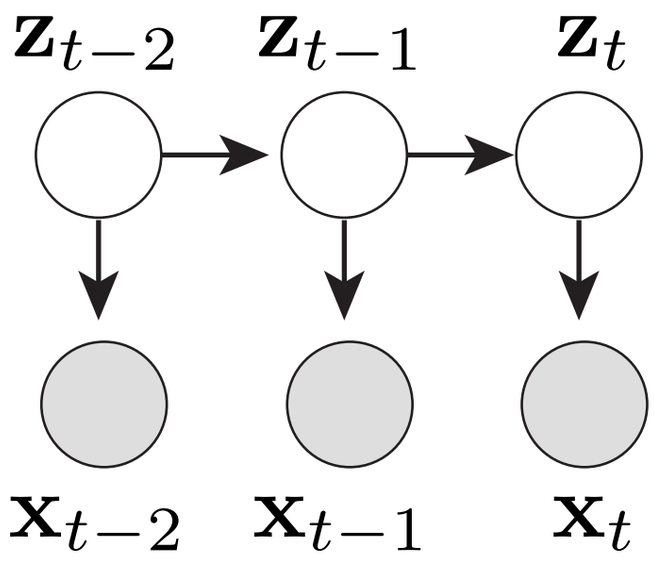
Latent state models



d-dimensional

n-dimensional

Latent state models



d-dimensional

n-dimensional

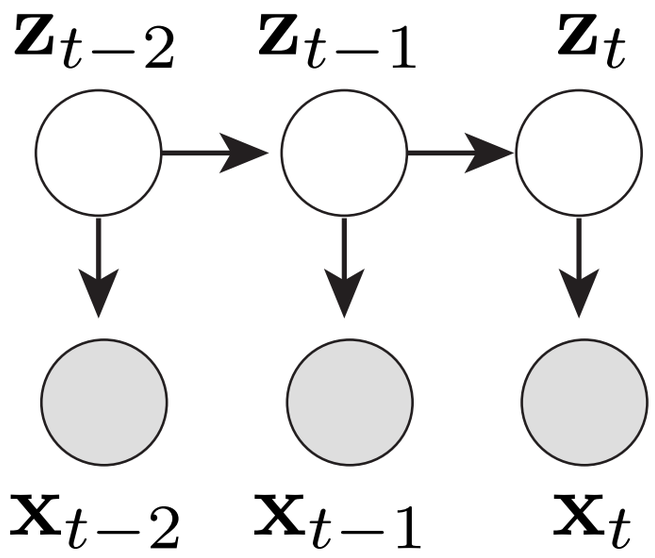
$$\mathbf{z}_t = \mathbf{A}\mathbf{z}_{t-1} + \mathbf{w}_t$$

$$\mathbf{x}_t = \mathbf{C}\mathbf{z}_t + \mathbf{v}_t$$

Where the noise terms are iid Gaussian:

$$\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q})$$

Latent state models



d-dimensional

n-dimensional

$$\mathbf{z}_t = \mathbf{A}\mathbf{z}_{t-1} + \mathbf{w}_t$$

$$\mathbf{x}_t = \mathbf{C}\mathbf{z}_t + \mathbf{v}_t$$

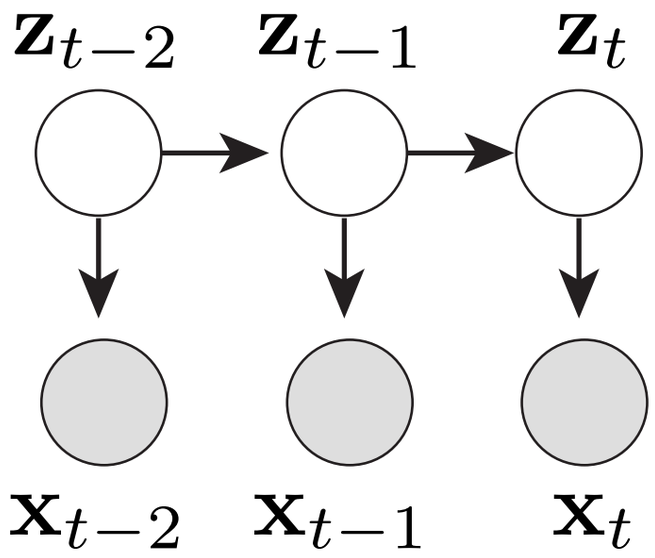
Where the noise terms are iid Gaussian:

$$\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q})$$

$$\mathbf{v}_t \sim \mathcal{N}(0, \mathbf{R})$$

$$\mathbf{z}_0 \sim \mathcal{N}(\mu_0, \Sigma)$$

Latent state models



d-dimensional

n-dimensional

$$\mathbf{z}_t = \mathbf{A}\mathbf{z}_{t-1} + \mathbf{w}_t$$
$$\mathbf{x}_t = \mathbf{C}\mathbf{z}_t + \mathbf{v}_t$$

Where the noise terms are iid Gaussian:

$$\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q})$$
$$\mathbf{v}_t \sim \mathcal{N}(0, \mathbf{R})$$
$$\mathbf{z}_0 \sim \mathcal{N}(\mu_0, \Sigma)$$

We are still in a happy linear gaussian world!

The noise terms are essential elements of the model : without **w**, deterministic behavior given by leading eigenvalue of **A**; without **v**, no latents

Without loss of generality, we can take **Q** to be the identity*, **R** can be any positive definite matrix

*If Q is not diagonal, it is still positive definite, so it can be diagonalized $\mathbf{Q} = \mathbf{E}\mathbf{D}\mathbf{E}^{\mathbf{T}}$

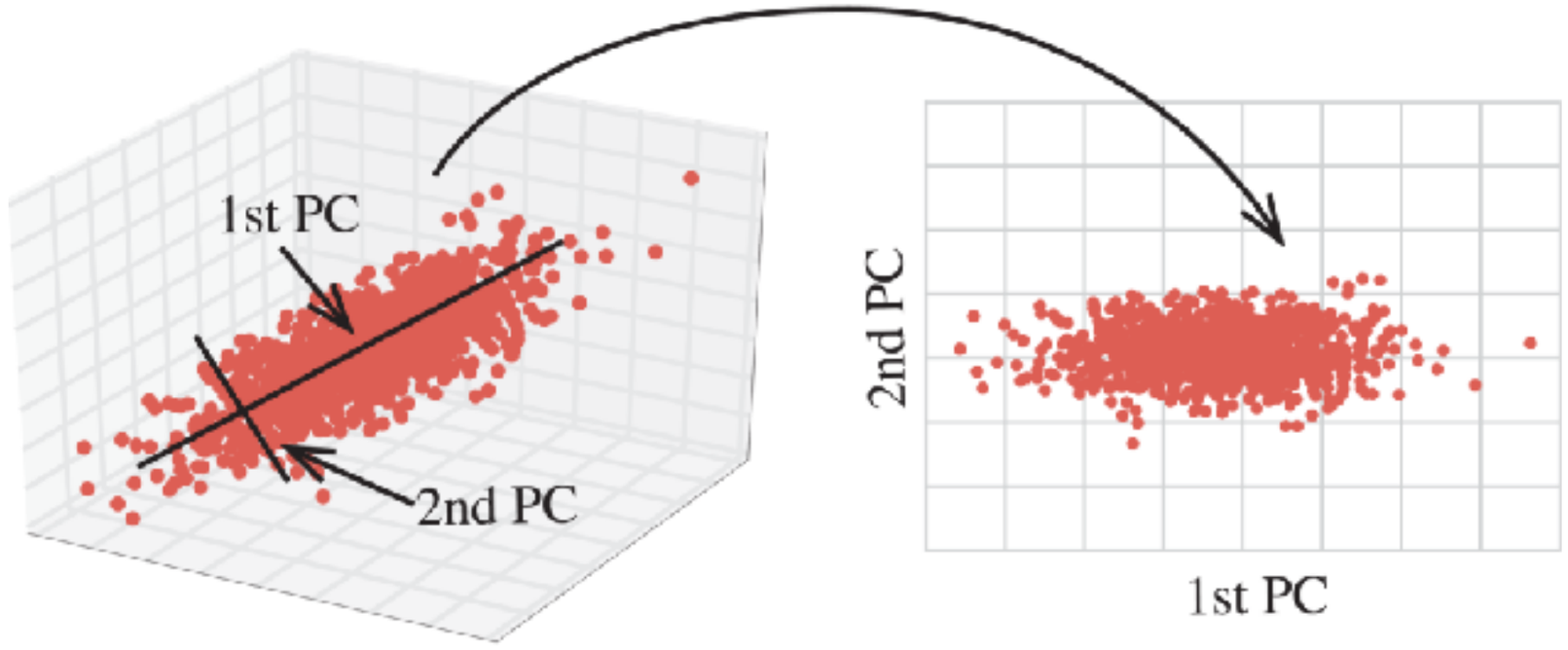
With the variable change (rotating axes) we can always rewrite this so that the noise term is the identity $\mathbf{y} = \mathbf{D}^{-1/2}\mathbf{E}^{\mathbf{T}}\mathbf{x}$

Check this at home!

Application : dimensionality reduction

Often we have d < n, i.e. the data itself is low-dimensional.

Can think about this as a time-dependent version of PCA
(we'll make this link formal later in the course)



Play with this next week in the lab!

LDS inference

We are given the data up to *s*, $\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_s\}$.

We want to estimate the latent state, $\mathbf{z}_t$.

If t>s this is prediction, if s=t **filtering**, if t<s **smoothing.**

LDS inference

We are given the data up to *s*, $\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_s\}$.

We want to estimate the latent state, $\mathbf{z}_t$.

If t>s this is prediction, if s=t **filtering**, if t<s **smoothing.**

$$\mathbf{z}_t|\mathbf{z}_{t-1} \sim \mathcal{N}(\mathbf{A}\mathbf{z}_{t-1}; \mathbf{Q})$$
$$\mathbf{x}_t|\mathbf{z}_t \sim \mathcal{N}(\mathbf{C}\mathbf{z}_t; \mathbf{R})$$

$$\mathrm{P}(\mathbf{z}_*, \mathbf{x}_*) = \mathrm{P}(\mathbf{z}_*)\prod_i \mathrm{P}(\mathbf{x}_i|\mathbf{z}_i)$$

* marks full series

$$\mathrm{P}(\mathbf{z}_*) = \mathrm{P}(\mathbf{z}_0)\prod_i \mathrm{P}(\mathbf{z}_t|\mathbf{z}_{t-1})$$

LDS inference

We are given the data up to *s*, $\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_s\}$.

We want to estimate the latent state, $\mathbf{z}_t$.

If t>s this is prediction, if s=t **filtering**, if t<s **smoothing.**

$$\mathbf{z}_t|\mathbf{z}_{t-1} \sim \mathcal{N}(\mathbf{A}\mathbf{z}_{t-1}; \mathbf{Q})$$
$$\mathbf{x}_t|\mathbf{z}_t \sim \mathcal{N}(\mathbf{C}\mathbf{z}_t; \mathbf{R})$$

$$\mathrm{P}(\mathbf{z}_*, \mathbf{x}_*) = \mathrm{P}(\mathbf{z}_*) \prod_i \mathrm{P}(\mathbf{x}_i|\mathbf{z}_i)$$

* marks full series

$$\mathrm{P}(\mathbf{z}_*) = \mathrm{P}(\mathbf{z}_0) \prod_i \mathrm{P}(\mathbf{z}_t|\mathbf{z}_{t-1})$$

Everything is Gaussian, so we could explicitly write it down.

LDS inference

We are given the data up to *s*, ***{x*$_1$*,…,x*$_s$*}.***

We want to estimate the latent state, $\mathbf{z}_t$.

If t>s this is prediction, if s=t **filtering**, if t<s **smoothing.**

$$\mathbf{z}_t|\mathbf{z}_{t-1} \sim \mathcal{N}(\mathbf{A}\mathbf{z}_{t-1}; \mathbf{Q})$$
$$\mathbf{x}_t|\mathbf{z}_t \sim \mathcal{N}(\mathbf{C}\mathbf{z}_t; \mathbf{R})$$

$$\mathrm{P}(\mathbf{z}_*, \mathbf{x}_*) = \mathrm{P}(\mathbf{z}_*) \prod_i \mathrm{P}(\mathbf{x}_i|\mathbf{z}_i)$$ * marks full series
$$\mathrm{P}(\mathbf{z}_*) = \mathrm{P}(\mathbf{z}_0) \prod_i \mathrm{P}(\mathbf{z}_t|\mathbf{z}_{t-1})$$

Everything is Gaussian, so we could explicitly write it down.

Kalman filtering uses recursion to compute things more efficiently.

LDS inference

We are given the data up to *s*, $\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_s\}$.

We want to estimate the latent state, $\mathbf{z}_t$.

If t>s this is prediction, if s=t **filtering**, if t<s **smoothing.**

$$\mathbf{z}_t|\mathbf{z}_{t-1} \sim \mathcal{N}(\mathbf{A}\mathbf{z}_{t-1}; \mathbf{Q})$$
$$\mathbf{x}_t|\mathbf{z}_t \sim \mathcal{N}(\mathbf{C}\mathbf{z}_t; \mathbf{R})$$

$$\mathrm{P}(\mathbf{z}_*, \mathbf{x}_*) = \mathrm{P}(\mathbf{z}_*) \prod_i \mathrm{P}(\mathbf{x}_i|\mathbf{z}_i)$$

* marks full series

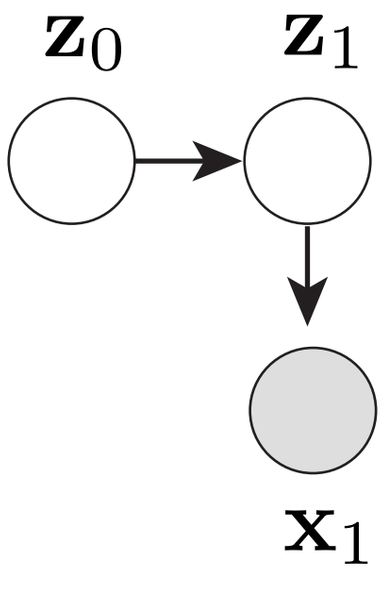
$$\mathrm{P}(\mathbf{z}_*) = \mathrm{P}(\mathbf{z}_0) \prod_i \mathrm{P}(\mathbf{z}_t|\mathbf{z}_{t-1})$$

Everything is Gaussian, so we could explicitly write it down.

Kalman filtering uses recursion to compute things more efficiently.

Note: This can be derived as a specific instance of message passing in trees (see e.g. Bishop, chp.6), but we're not going to do this here.

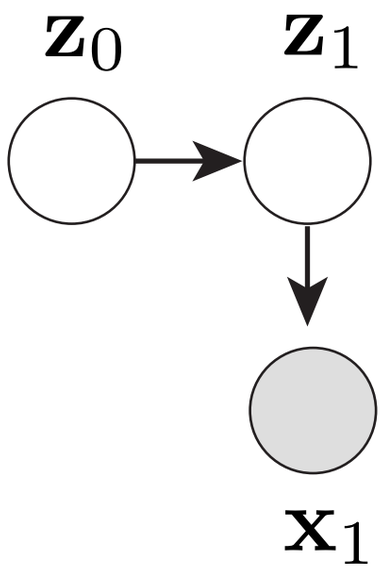
Intuitively, we can construct the forward pass by induction



2 sub-steps:

$\mathbf{z}_1 \big| \mathbf{z}_0$ **Take into account what we know about z_0**

Intuitively, we can construct the forward pass by induction



2 sub-steps:

$\mathbf{z}_1 \big| \mathbf{z}_0$ **Take into account what we know about z_0**

$\mathbf{z}_1 \big| \mathbf{x}_1, \mathbf{z}_0$ **Take into account evidence y_1**

Basic Gaussian identities

a d-dimensional multidimensional gaussian (normal) density for $\mathbf{x}$ is:

$$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-d/2}|\boldsymbol{\Sigma}|^{-1/2}\exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]$$

Rescaling and shifting gaussian variables

$$\mathrm{E}[\mathbf{A}\mathbf{x}+\mathbf{y}] = \mathbf{A}(\mathrm{E}[\mathbf{x}])+\mathbf{y}$$
$$\mathrm{Covar}[\mathbf{A}\mathbf{x}+\mathbf{y}] = \mathbf{A}(\mathrm{Covar}[\mathbf{x}])\mathbf{A}^T$$

Sum of 2 independent variables **c** = **a**+**b** (Bienaymé)

$$\Sigma_c = \Sigma_a + \Sigma_b$$

Product of gaussians is another (unnormalized) gaussian

$$\mathcal{N}(\mathbf{a}, \mathbf{A}) \cdot \mathcal{N}(\mathbf{b}, \mathbf{B}) \propto \mathcal{N}(\mathbf{c}, \mathbf{C})$$

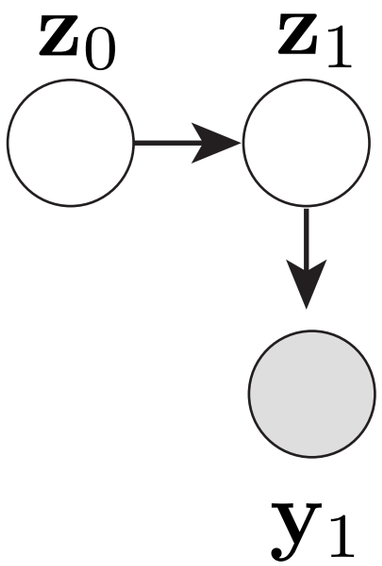
$$\mathbf{C} = \left(\mathbf{A}^{-1} + \mathbf{B}^{-1}\right)^{-1}$$

$$\mathbf{c} = \mathbf{C}\mathbf{A}^{-1}\mathbf{a} + \mathbf{C}\mathbf{B}^{-1}\mathbf{b}$$

Note: the normalizing constant is also normal in **a** and **b**

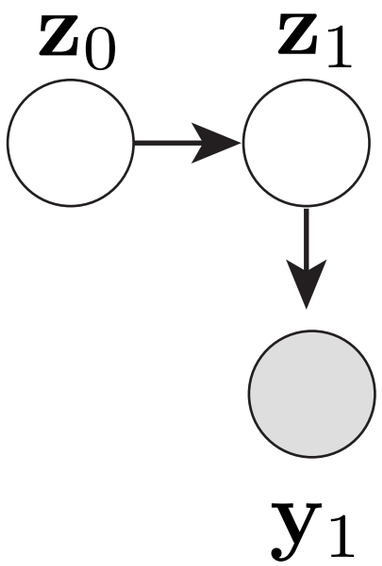
$$z_c = (2\pi)^{-d/2}|\mathbf{C}|^{+1/2}|\mathbf{A}|^{-1/2}|\mathbf{B}|^{-1/2}\exp\left[-\frac{1}{2}(\mathbf{a}^T\mathbf{A}^{-1}\mathbf{a} + \mathbf{b}^T\mathbf{B}^{-1}\mathbf{b} - \mathbf{c}^T\mathbf{C}^{-1}\mathbf{c})\right]$$

Intuitively, we can construct the forward pass by induction



2 sub-steps:

Intuitively, we can construct the forward pass by induction



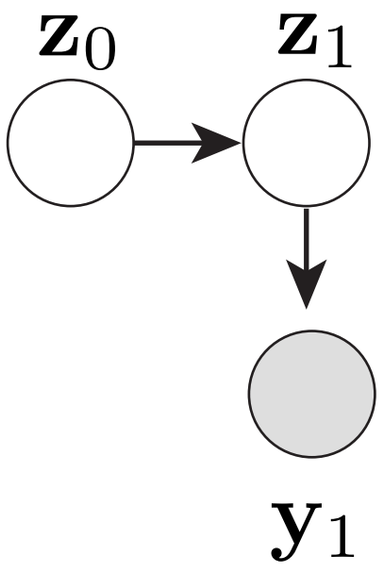
2 sub-steps:

Forecasting

$$\mathbf{z}_1|\mathbf{z}_0 \qquad \mu_{1|0} = A\mu_0$$

$$\Sigma_{1|0} = A\Sigma_0 A^{\mathrm{T}} + Q$$

Intuitively, we can construct the forward pass by induction



2 sub-steps:

Forecasting

$$\mathbf{z}_1|\mathbf{z}_0 \qquad \mu_{1|0} = A\mu_0$$

$$\Sigma_{1|0} = A\Sigma_0 A^{\mathrm{T}} + Q$$

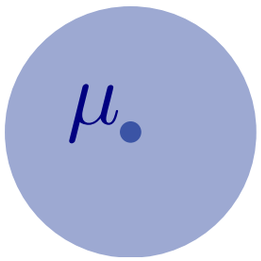
Combine with new evidence

$$\mathbf{z}_1|\mathbf{x}_1,\, \mathbf{z}_0 \qquad \mu_{1|1} = \mu_{1|0} + \mathbf{K}(\mathbf{x}_1 - \mathbf{C}\mu_{1|0})$$

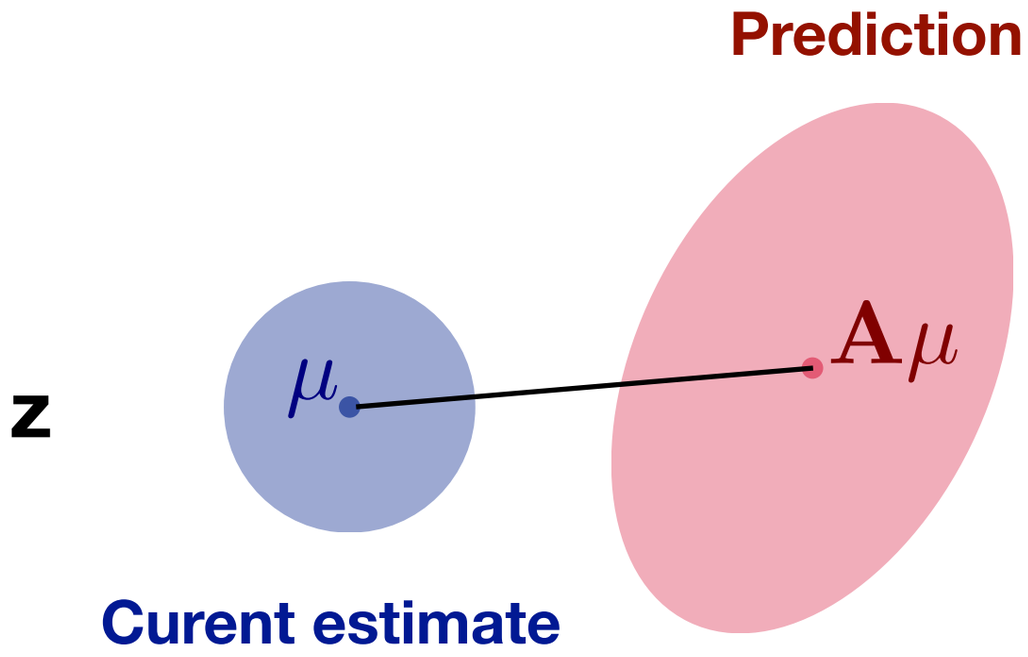
$$\Sigma_{1|1} = \Sigma_{1|0} - \mathbf{K}\mathbf{C}\Sigma_{1|0}$$

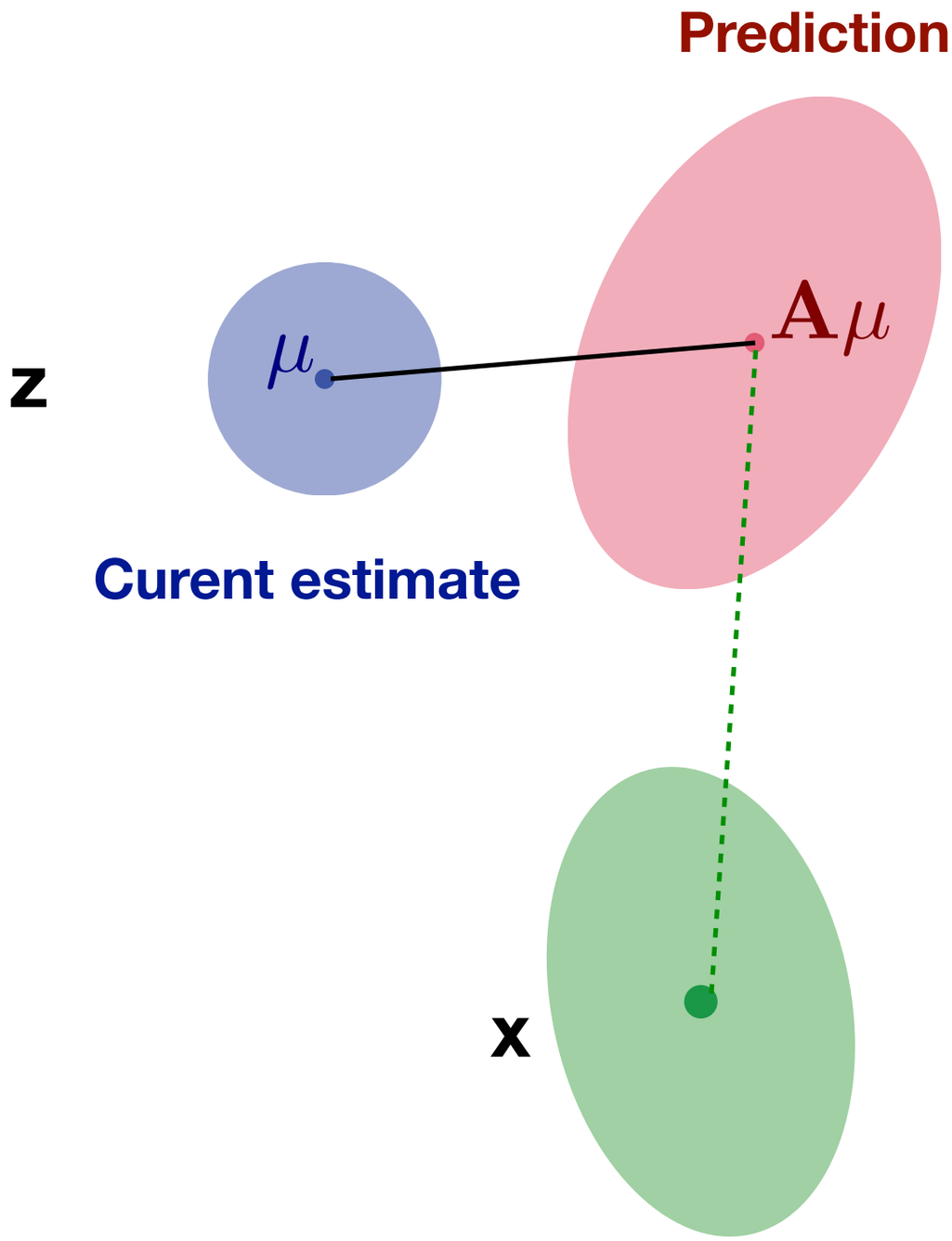
$$\mathbf{K} = \Sigma_{1|0}\mathbf{C}^{\mathrm{T}}(\mathbf{C}\Sigma_{1|0}\mathbf{C}^{\mathrm{T}} + R)^{-1}$$

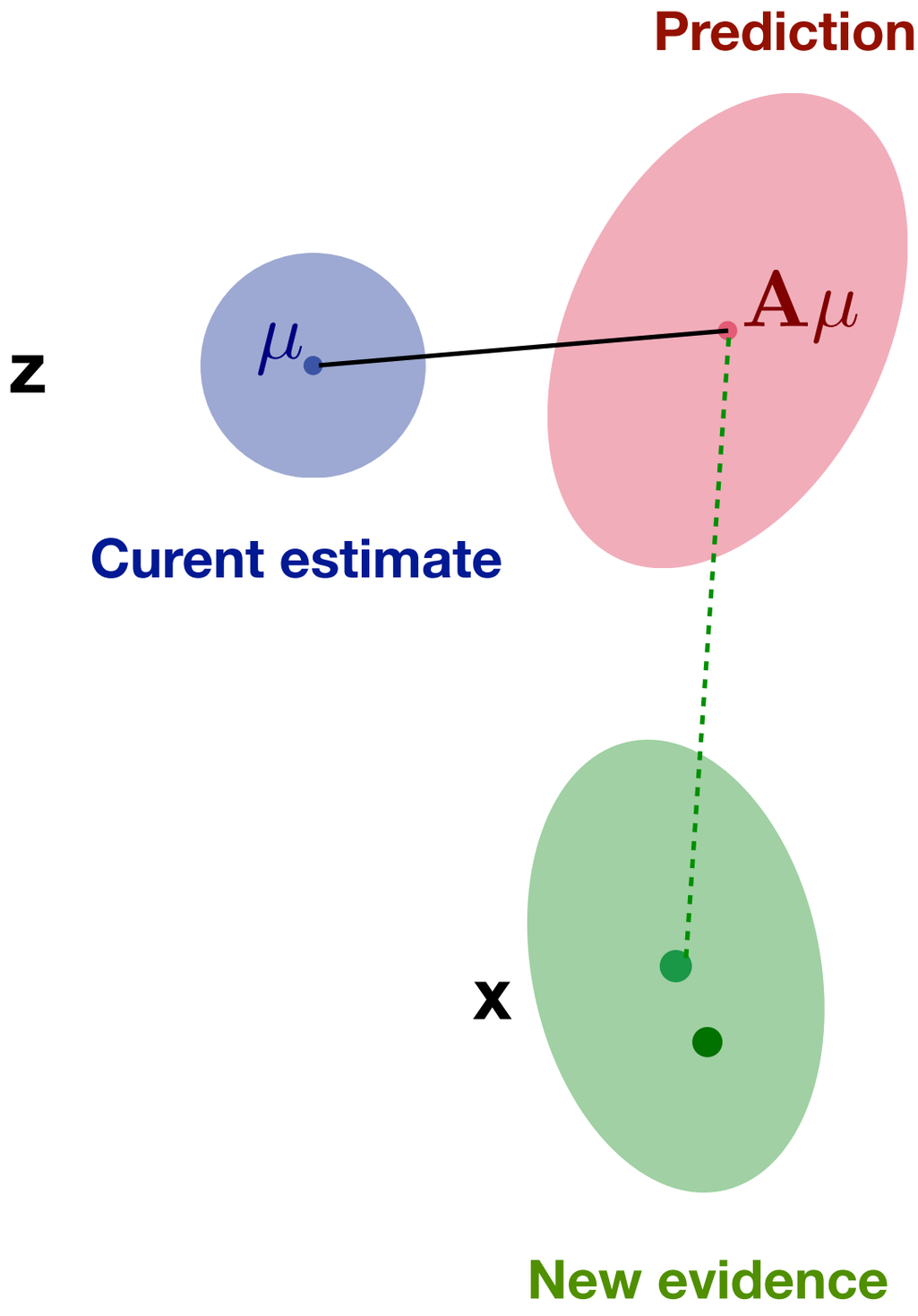
z

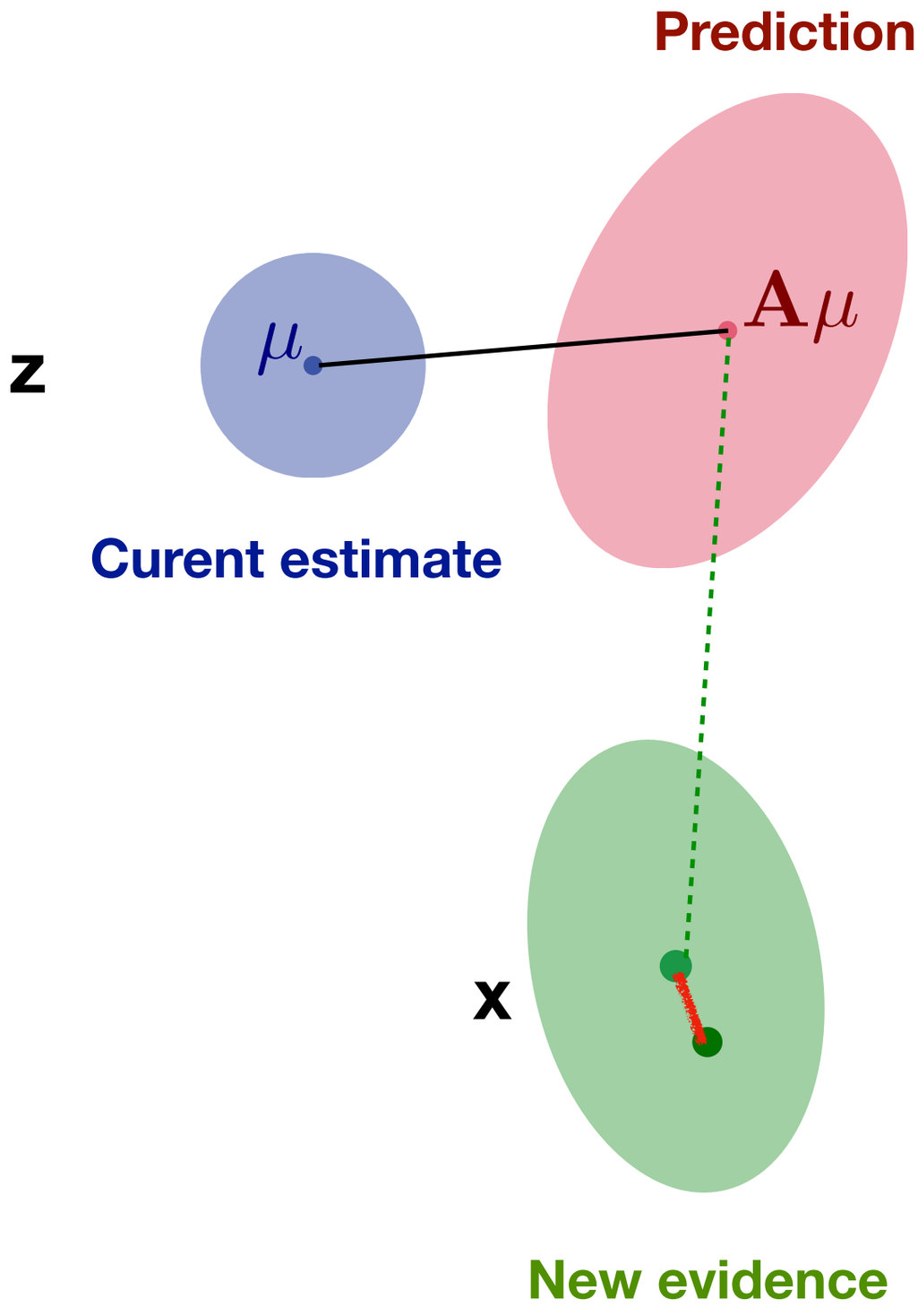


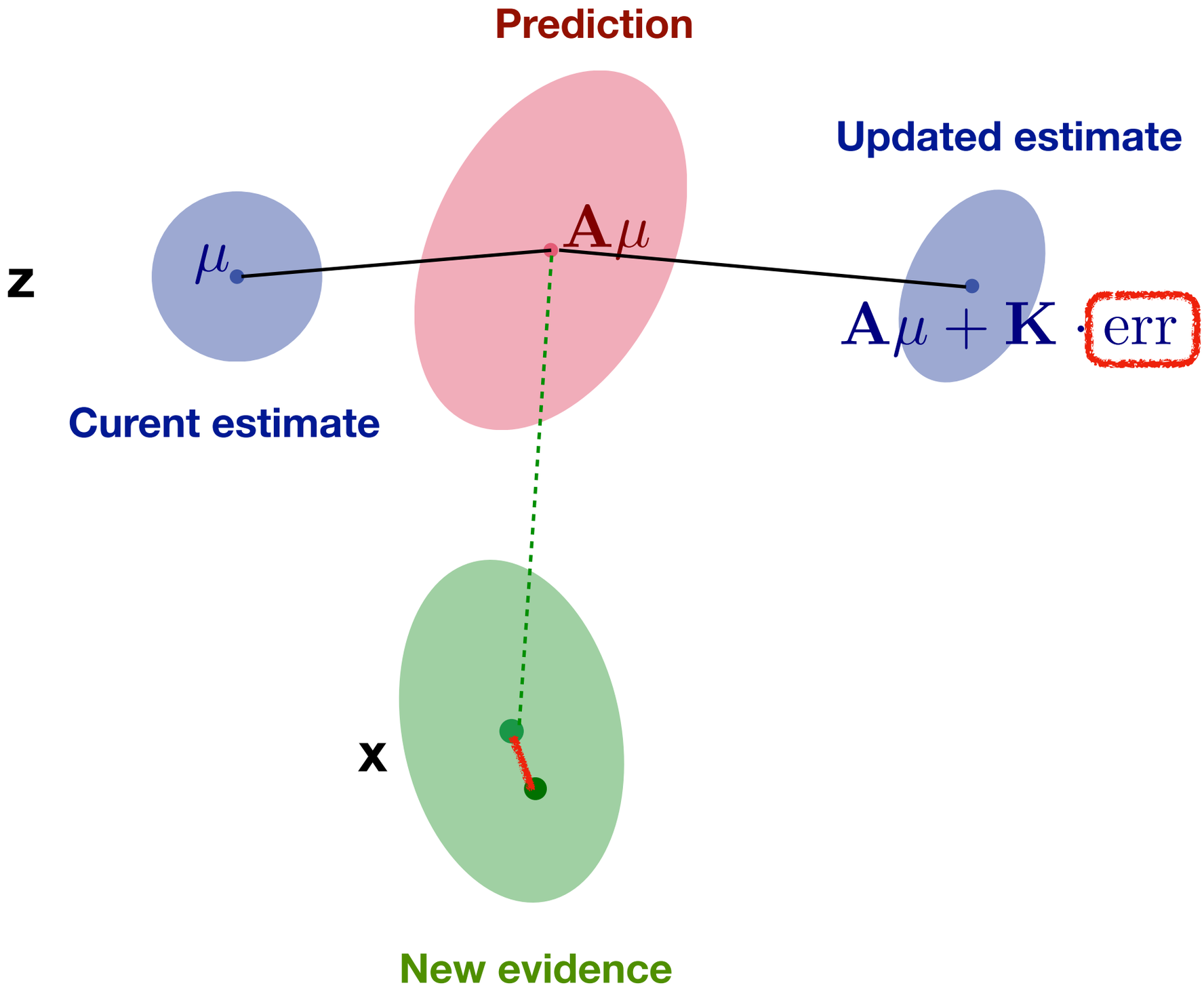
Curent estimate

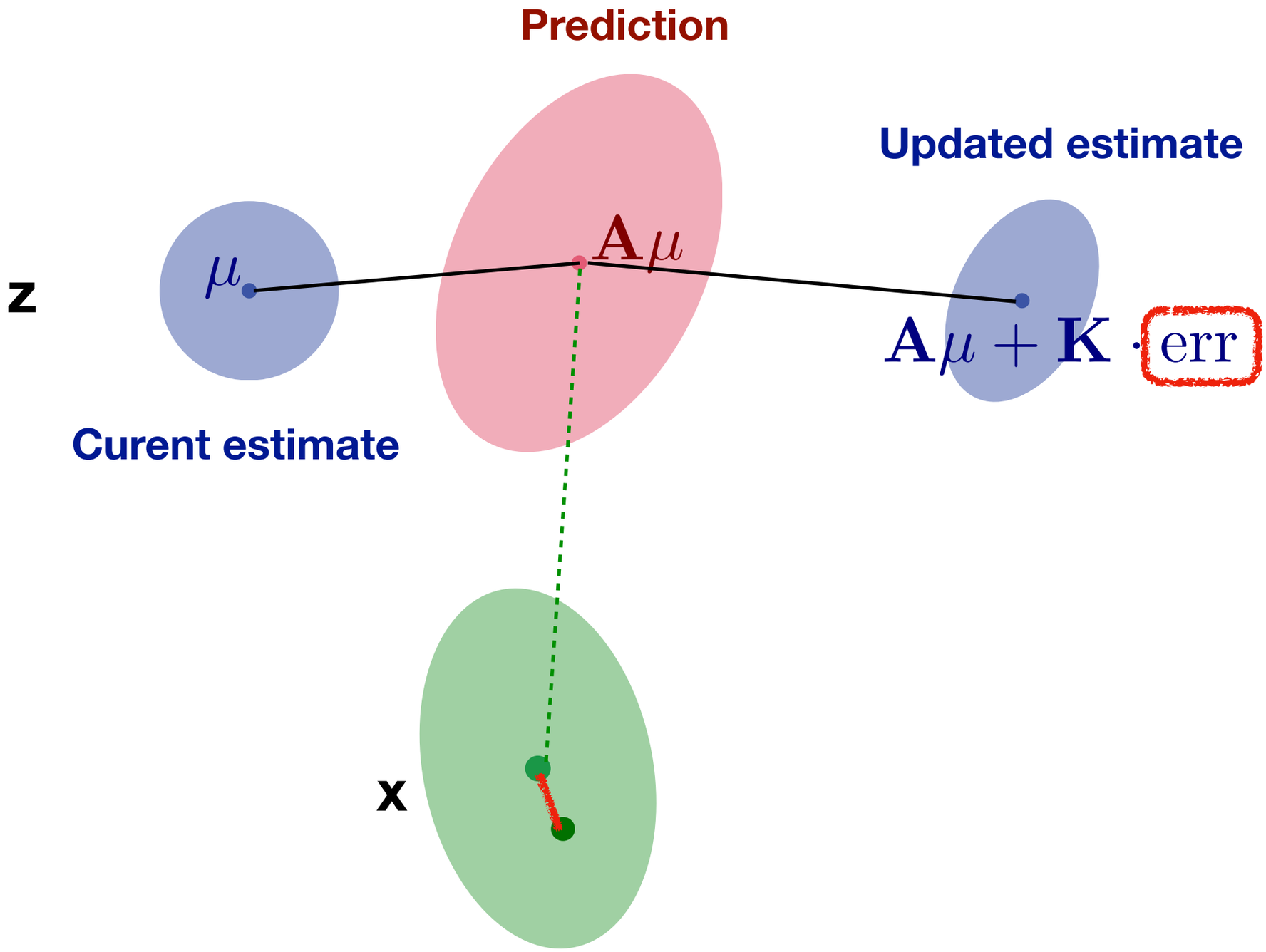

Prediction
z
μ
$\mathbf{A}\mu$
Curent estimate


Prediction
μ
$\mathbf{A}\mu$
z
Curent estimate
x

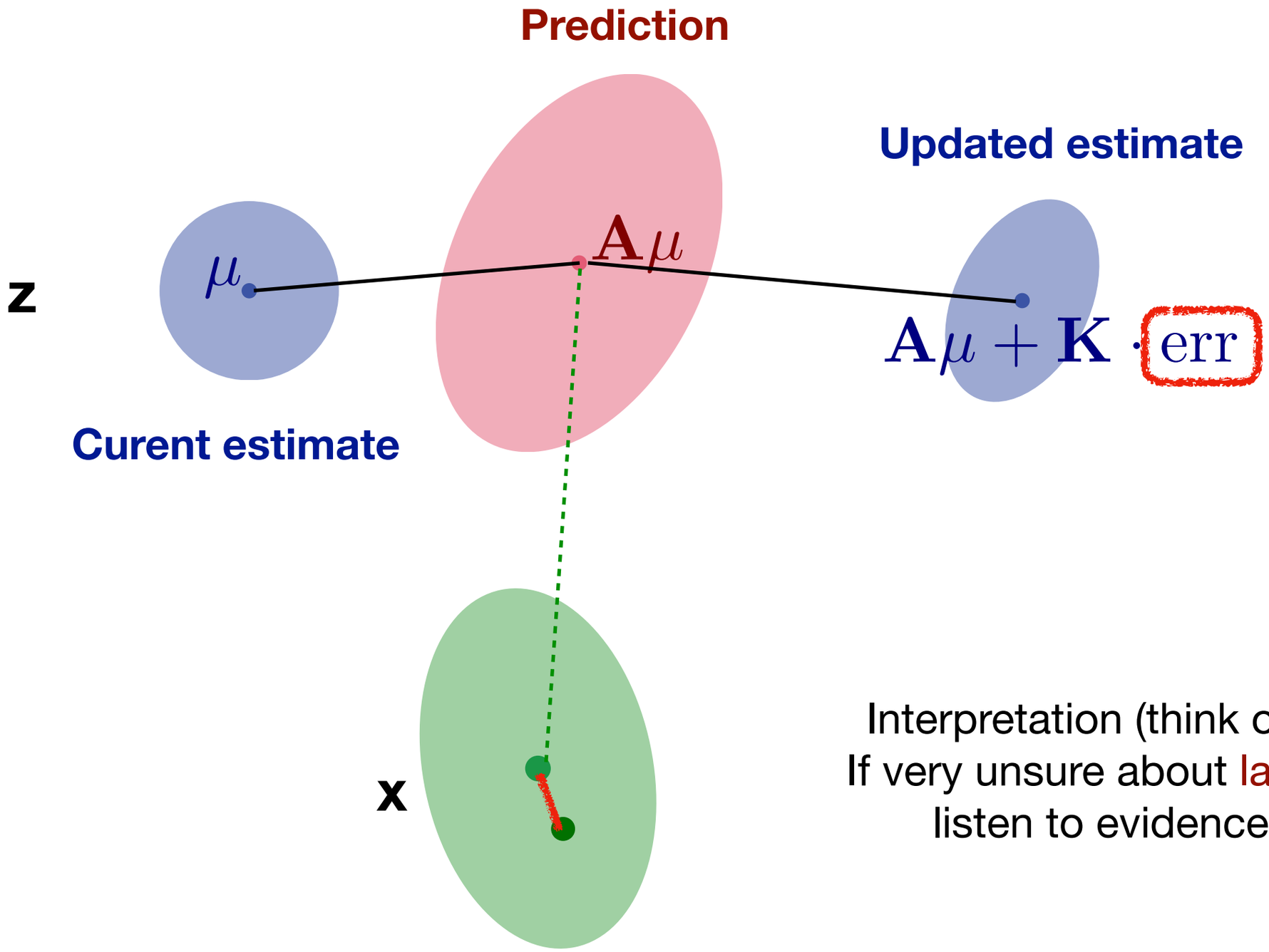

Prediction
μ
$\mathbf{A}\mu$
z
Curent estimate
x
New evidence


Prediction
μ
$\mathbf{A}\mu$
z
Curent estimate
x
New evidence


Prediction
Updated estimate
z
μ
$\mathbf{A}\mu$
$\mathbf{A}\mu + \mathbf{K} \cdot \mathrm{err}$
Curent estimate
x
New evidence

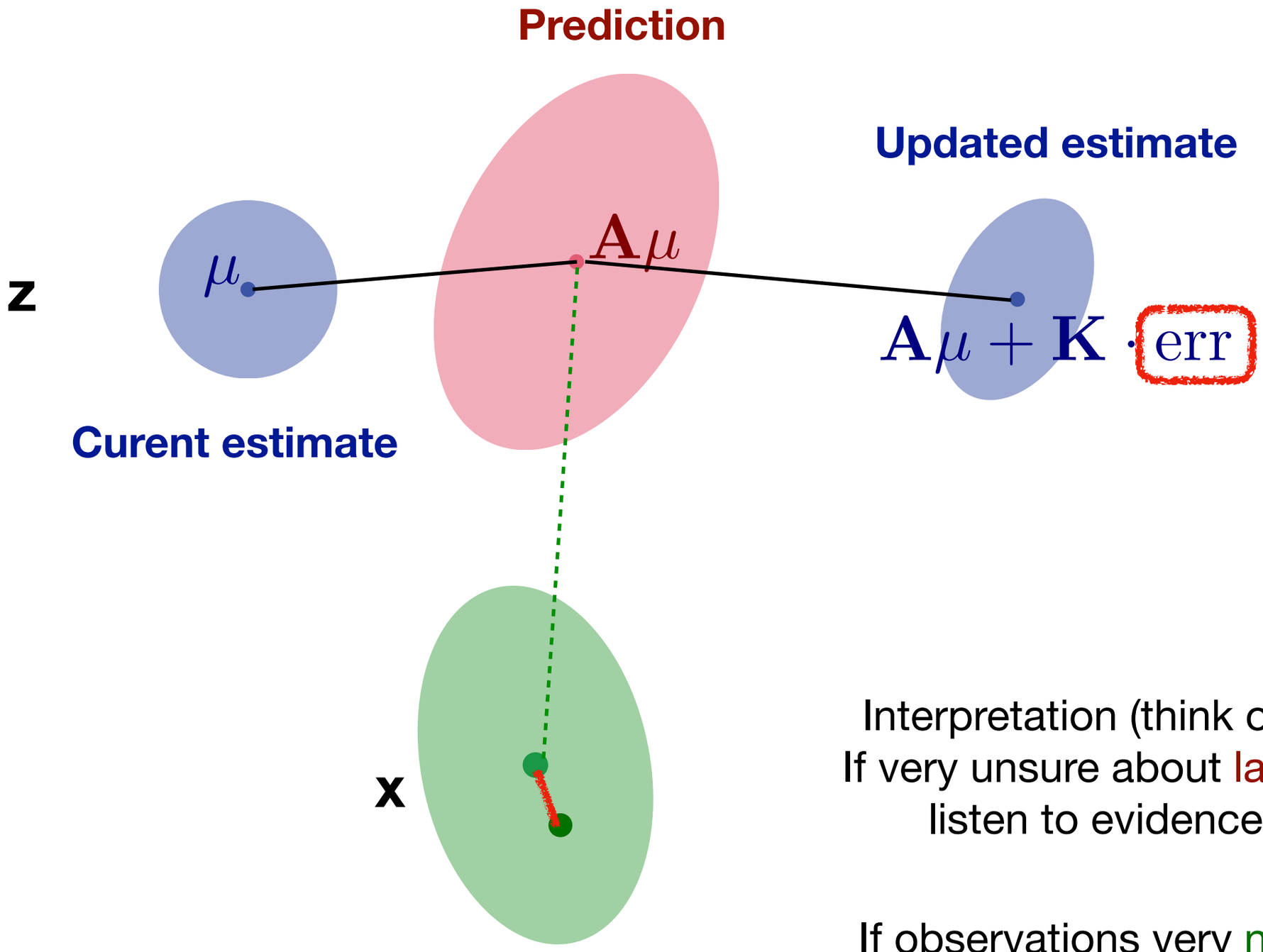


$$\mathbf{K} = \Sigma_{1|0}\mathbf{C}^{\mathrm{T}}(\mathbf{C}\Sigma_{1|0}\mathbf{C}^{\mathrm{T}} + R)^{-1}$$



Interpretation (think of scales):
If very unsure about latent state,
listen to evidence more

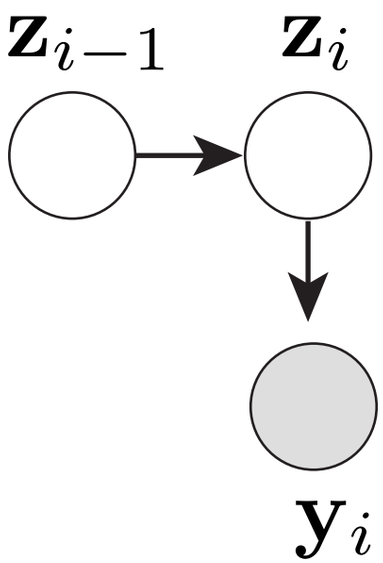
$$\mathbf{K} = \Sigma_{1|0}\mathbf{C}^{\mathrm{T}}(\mathbf{C}\Sigma_{1|0}\mathbf{C}^{\mathrm{T}} + R)^{-1}$$



Interpretation (think of scales):
If very unsure about latent state,
listen to evidence more

If observations very nosy (R large),
listen to evidence less

$$\mathbf{K} = \Sigma_{1|0}\mathbf{C}^{\mathrm{T}}(\mathbf{C}\Sigma_{1|0}\mathbf{C}^{\mathrm{T}} + R)^{-1}$$

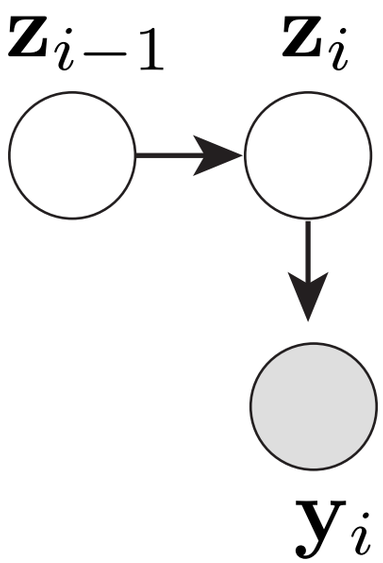


Exactly in the same way, if we have a posterior distribution for $\mathbf{z}_{i-1}$, we can use it instead of $P(\mathbf{z}_0)$ to compute the posterior for $\mathbf{z}_i$.

$$\mu_{i|i} = \mu_{i|i-1} + \mathbf{K}_i(\mathbf{x}_i - \mathbf{C}\mu_{i|i-1})$$

$$\Sigma_{i|i} = \Sigma_{i|i-1} - \mathbf{K}_i\mathbf{C}\Sigma_{i|i-1}$$

Notation: double index i|j:
i-we are estimating variable $\mathbf{z}_i$,
j-datapoints observed up to (including) $\mathbf{y}_j$



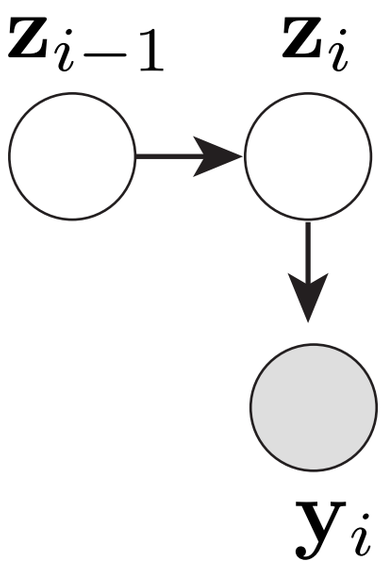
Exactly in the same way, if we have a posterior distribution for $\mathbf{z}_{i-1}$, we can use it instead of $P(\mathbf{z}_0)$ to compute the posterior for $\mathbf{z}_i$.

$$\mu_{i|i} = \mu_{i|i-1} + \mathbf{K}_i(\mathbf{x}_i - \mathbf{C}\mu_{i|i-1})$$

$$\Sigma_{i|i} = \Sigma_{i|i-1} - \mathbf{K}_i\mathbf{C}\Sigma_{i|i-1}$$

$$\mathbf{K}_i = \Sigma_{i|i-1}\mathbf{C}^{\mathrm{T}}(\mathbf{C}\Sigma_{i|i-1}\mathbf{C}^{\mathrm{T}} + R)^{-1}$$

Notation: double index i|j:
i-we are estimating variable $\mathbf{z}_i$,
j-datapoints observed up to (including) $\mathbf{y}_j$



Exactly in the same way, if we have a posterior distribution for $\mathbf{z}_{i-1}$, we can use it instead of $P(\mathbf{z}_0)$ to compute the posterior for $\mathbf{z}_i$.

$$\mu_{i|i} = \mu_{i|i-1} + \mathbf{K}_i(\mathbf{x}_i - \mathbf{C}\mu_{i|i-1})$$

$$\Sigma_{i|i} = \Sigma_{i|i-1} - \mathbf{K}_i\mathbf{C}\Sigma_{i|i-1}$$

$$\mathbf{K}_i = \Sigma_{i|i-1}\mathbf{C}^{\mathrm{T}}(\mathbf{C}\Sigma_{i|i-1}\mathbf{C}^{\mathrm{T}} + R)^{-1}$$ **Kalman gain**

The gain changes dynamically!

Notation: double index i|j:
i-we are estimating variable $\mathbf{z}_i$,
j-datapoints observed up to (including) $\mathbf{y}_j$

Smoothing - backward sweep

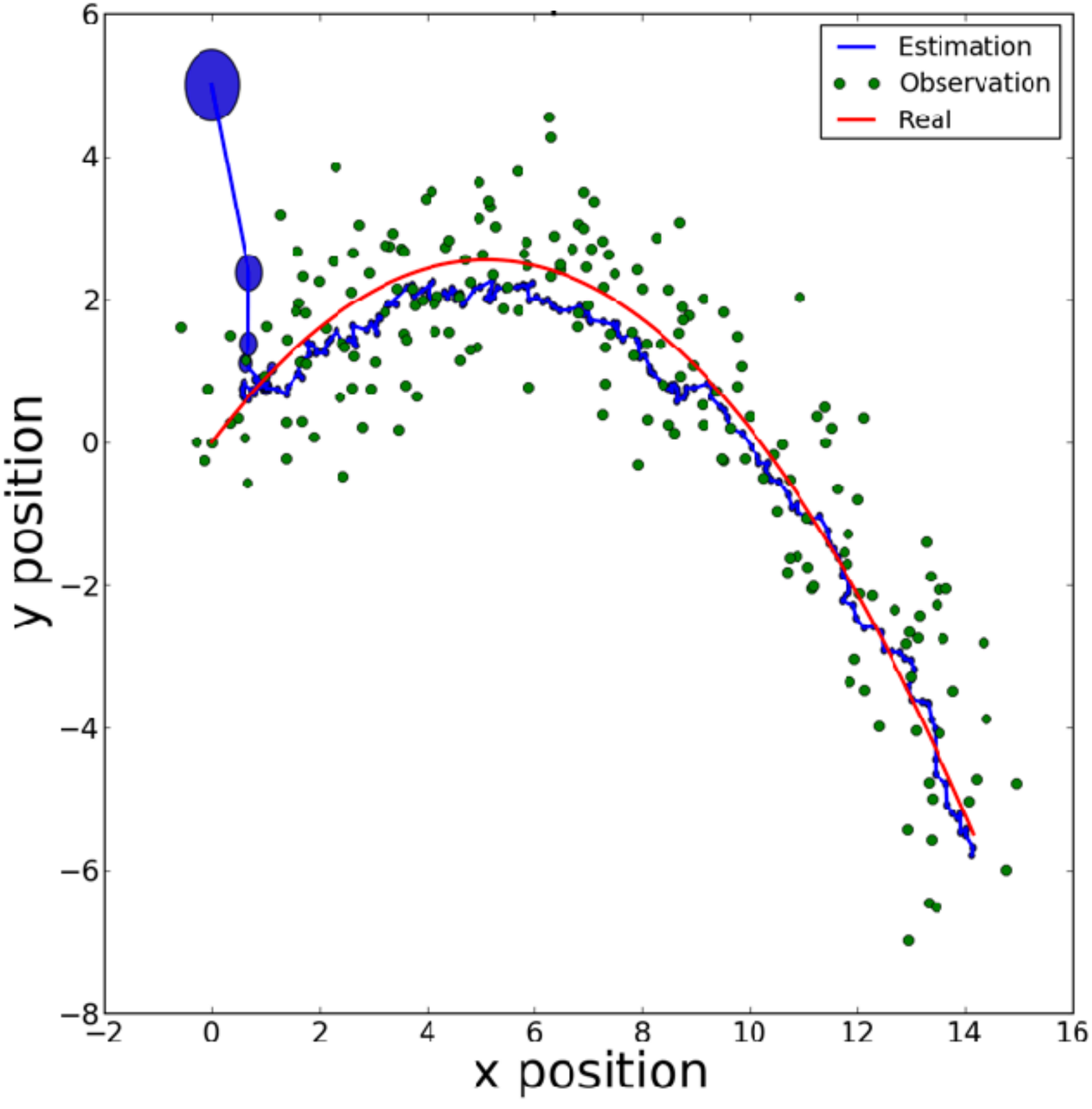
Same principle, but now we propagate information backwards in time, using the estimates we have already (we no longer need the data)

$$\mu_{i|t} = \mu_{i|i} + \mathbf{F}_i(\mu_{i+1|t} - \mu_{i+1|i})$$

$$\Sigma_{i|t} = \mathbf{F}_i(\Sigma_{i+1|t} - \Sigma_{i+1|i})\mathbf{F}_i^{\mathrm{T}} + \Sigma_{i|i}$$

$$\mathbf{F}_i = \Sigma_{i|i}\mathbf{A}^{\mathrm{T}}\Sigma_{i+1|i}^{-1}$$

If you want to prove these, you can check one variant in tsa4.pdf p299-300
The notation is going to be slightly different…


Estimation
Observation
Real
y position
x position
6
4
2
0
−2
−4
−6
−8
−2
0
2
4
6
8
10
12
14
16

Estimating parameters via expectation maximization

Find parameters that are most consistent with the observed data

To simplify notation we use shorthand $\theta = \{\mathbf{A},\, \mathbf{C},\, \mathbf{Q},\, \mathbf{R}, \mu_0, \Sigma_0\}$

The goal is to find the parameters that maximize the (log) likelihood

$$\mathcal{L}(\theta) = \log \mathrm{P}(\mathbf{x}_* | \theta)$$

Estimating parameters via expectation maximization

Find parameters that are most consistent with the observed data

To simplify notation we use shorthand $\theta = \{\mathbf{A},\ \mathbf{C},\ \mathbf{Q},\ \mathbf{R}, \mu_0, \Sigma_0\}$

The goal is to find the parameters that maximize the (log) likelihood

$$\mathcal{L}(\theta) = \log \mathrm{P}(\mathbf{x}_*|\theta)$$

Which we get after marginalizing out the latents

$$\mathrm{P}(\mathbf{x}_*|\theta) = \int \mathrm{P}(\mathbf{x}_*, \mathbf{z}_*|\theta)\mathbf{dz}$$

General idea of EM

$$\log \int_z \mathrm{P}(\mathbf{x}, \mathbf{z}|\theta)\mathbf{dz} = \log \int_z Q(\mathbf{z}) \frac{\mathrm{P}(\mathbf{x},\mathbf{z}|\theta)}{Q(\mathbf{z})}\mathbf{dz}$$

General idea of EM

$$\log \int_z \mathrm{P}(\mathbf{x}, \mathbf{z}|\theta)\mathbf{dz} = \log \int_z Q(\mathbf{z}) \frac{\mathrm{P}(\mathbf{x},\mathbf{z}|\theta)}{Q(\mathbf{z})}\mathbf{dz}$$

$$\geq \int_z Q(\mathbf{z}) \log \frac{\mathrm{P}(\mathbf{x},\mathbf{z}|\theta)}{Q(\mathbf{z})}\mathbf{dz}$$

General idea of EM

$$\log \int_z \mathrm{P}(\mathbf{x}, \mathbf{z}|\theta)\mathbf{dz} = \log \int_z Q(\mathbf{z}) \frac{\mathrm{P}(\mathbf{x},\mathbf{z}|\theta)}{Q(\mathbf{z})}\mathbf{dz} \quad = \log\left(\mathbb{E}_Q\left[\frac{\mathrm{P}(\mathbf{x},\mathbf{z}|\theta)}{Q(\mathbf{z})}\right]\right)$$

via **Jensen's Inequality** $\geq \int_z Q(\mathbf{z}) \log \frac{\mathrm{P}(\mathbf{x},\mathbf{z}|\theta)}{Q(\mathbf{z})}\mathbf{dz}$ $= \mathbb{E}_Q\left[\log \frac{\mathrm{P}(\mathbf{x},\mathbf{z}|\theta)}{Q(\mathbf{z})}\right]$

Big picture: log of integral is not nice, so we replace it with bound on integral of log

General idea of EM

$$\log \int_z \mathrm{P}(\mathbf{x}, \mathbf{z}|\theta)\mathbf{dz} = \log \int_z Q(\mathbf{z}) \frac{\mathrm{P}(\mathbf{x},\mathbf{z}|\theta)}{Q(\mathbf{z})}\mathbf{dz} \quad = \log\left(\mathbb{E}_Q\left[\frac{\mathrm{P}(\mathbf{x},\mathbf{z}|\theta)}{Q(\mathbf{z})}\right]\right)$$

via **Jensen's Inequality** $\geq \int_z Q(\mathbf{z}) \log \frac{\mathrm{P}(\mathbf{x},\mathbf{z}|\theta)}{Q(\mathbf{z})}\mathbf{dz}$ $= \mathbb{E}_Q\left[\log \frac{\mathrm{P}(\mathbf{x},\mathbf{z}|\theta)}{Q(\mathbf{z})}\right]$

$$= \int_Q Q(\mathbf{z}) \log \mathrm{P}(\mathbf{x}, \mathbf{z}|\theta))\mathbf{dz} - \int_Q Q(\mathbf{z}) \log Q(\mathbf{z})\mathbf{dz}$$

Big picture: log of integral is not nice, so we replace it with bound on integral of log

General idea of EM

$$\log \int_z \mathrm{P}(\mathbf{x}, \mathbf{z}|\theta)\mathbf{dz} = \log \int_z Q(\mathbf{z}) \frac{\mathrm{P}(\mathbf{x},\mathbf{z}|\theta)}{Q(\mathbf{z})}\mathbf{dz} \quad = \log\left(\mathbb{E}_Q\left[\frac{\mathrm{P}(\mathbf{x},\mathbf{z}|\theta)}{Q(\mathbf{z})}\right]\right)$$

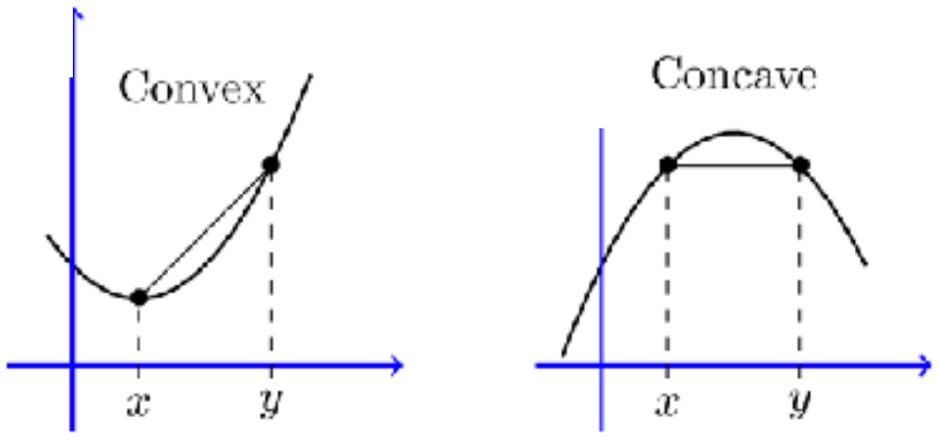
via **Jensen's Inequality** $\geq \int_z Q(\mathbf{z}) \log \frac{\mathrm{P}(\mathbf{x},\mathbf{z}|\theta)}{Q(\mathbf{z})}\mathbf{dz}$ $= \mathbb{E}_Q\left[\log \frac{\mathrm{P}(\mathbf{x},\mathbf{z}|\theta)}{Q(\mathbf{z})}\right]$

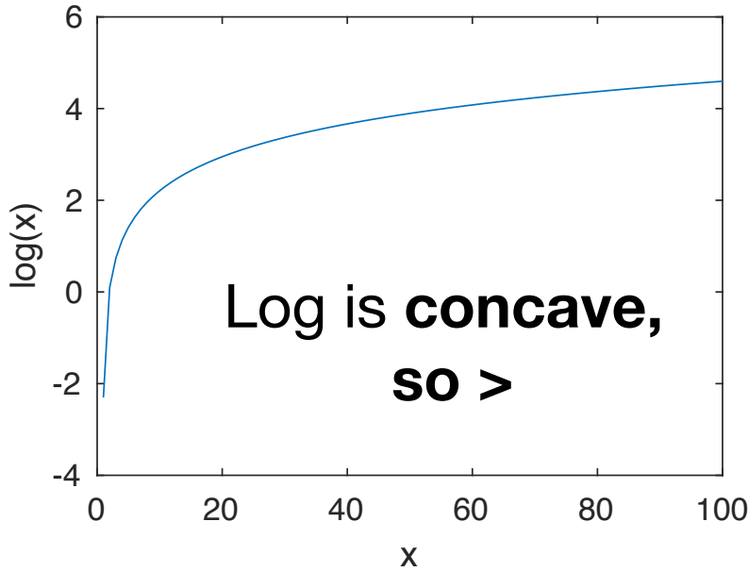
$$= \int_Q Q(\mathbf{z}) \log \mathrm{P}(\mathbf{x}, \mathbf{z}|\theta))\mathbf{dz} - \int_Q Q(\mathbf{z}) \log Q(\mathbf{z})\mathbf{dz}$$

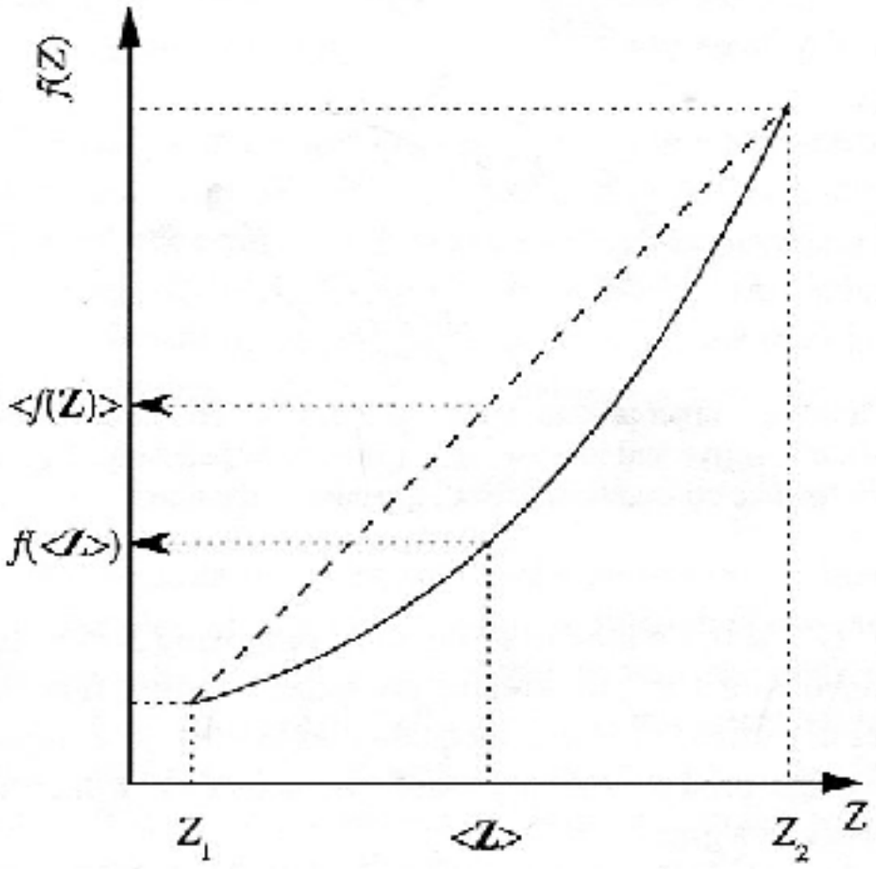
$$= \mathcal{F}(Q, \theta)$$ **Free energy**

Big picture: log of integral is not nice, so we replace it with bound on integral of log

Reminder: convex vs concave functions; Jensen's inequality







$$f(\mathbb{E}[x]) \leq \mathbb{E}[f(x)]$$

If f is **convex**

EM alternates between finding a good approximation Q,
and then changing the parameters to improve the approx likelihood

This is done coordinate-wise: improve one keeping the other fixed, then swap.

$$\textbf{E step:} \qquad Q_{k+1} \leftarrow \underset{Q}{\arg\max} \;\; \mathcal{F}(Q, \theta_k)$$

$$\textbf{M step:} \qquad \theta_{k+1} \leftarrow \underset{\theta}{\arg\max} \;\; \mathcal{F}(Q_{k+1}, \theta)$$

EM alternates between finding a good approximation Q,
and then changing the parameters to improve the approx likelihood

This is done coordinate-wise: improve one keeping the other fixed, then swap.

$$\textbf{E step:} \qquad Q_{k+1} \leftarrow \underset{Q}{\arg\max} \;\; \mathcal{F}(Q, \theta_k)$$

$$\textbf{M step:} \qquad \theta_{k+1} \leftarrow \underset{\theta}{\arg\max} \;\; \mathcal{F}(Q_{k+1}, \theta)$$

It is guaranteed that the likelihood will **never decrease** during this procedure.

EM alternates between finding a good approximation Q,
and then changing the parameters to improve the approx likelihood

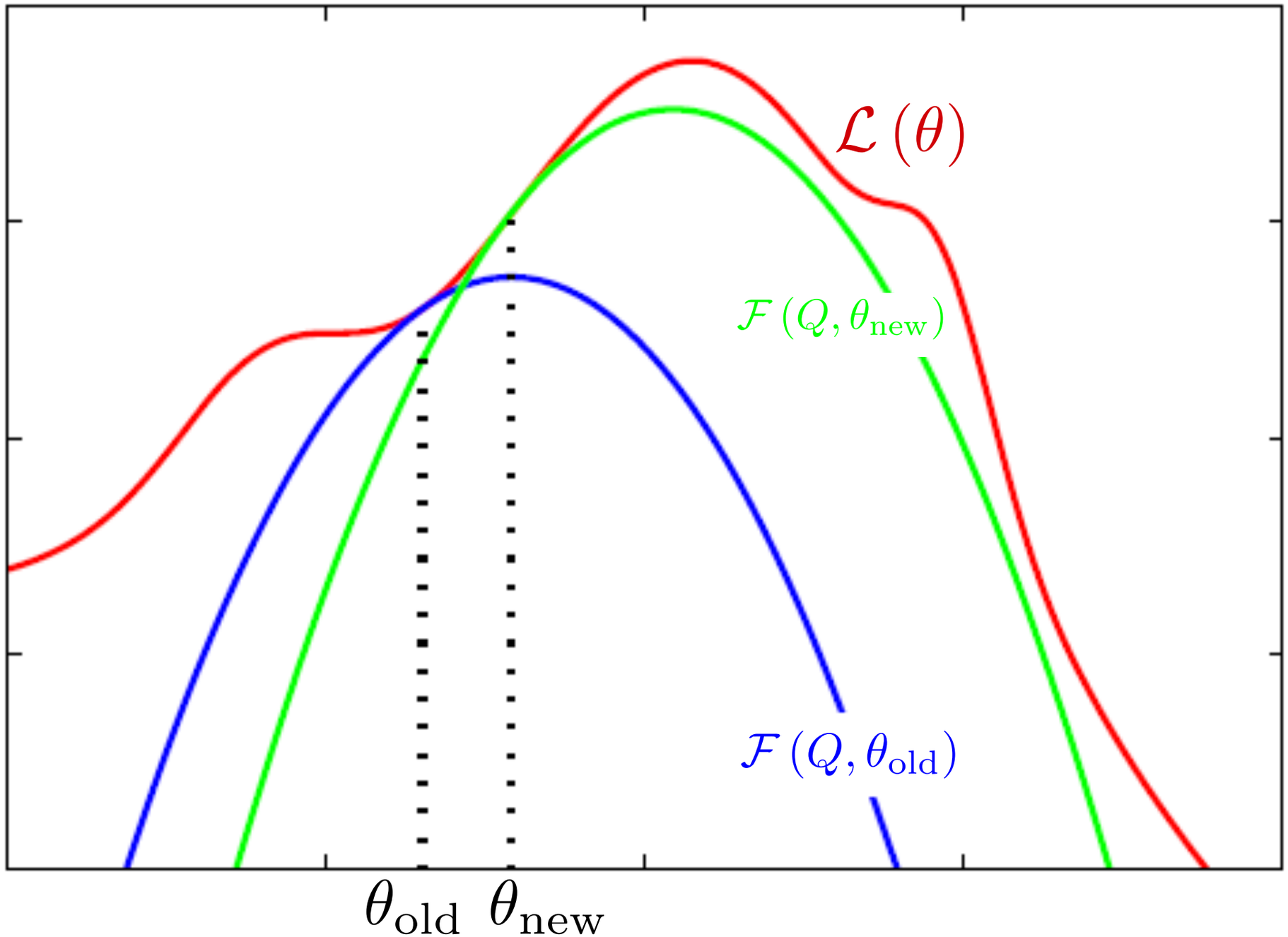
This is done coordinate-wise: improve one keeping the other fixed, then swap.

$$\textbf{E step:} \qquad Q_{k+1} \leftarrow \underset{Q}{\arg\max} \;\; \mathcal{F}(Q, \theta_k)$$

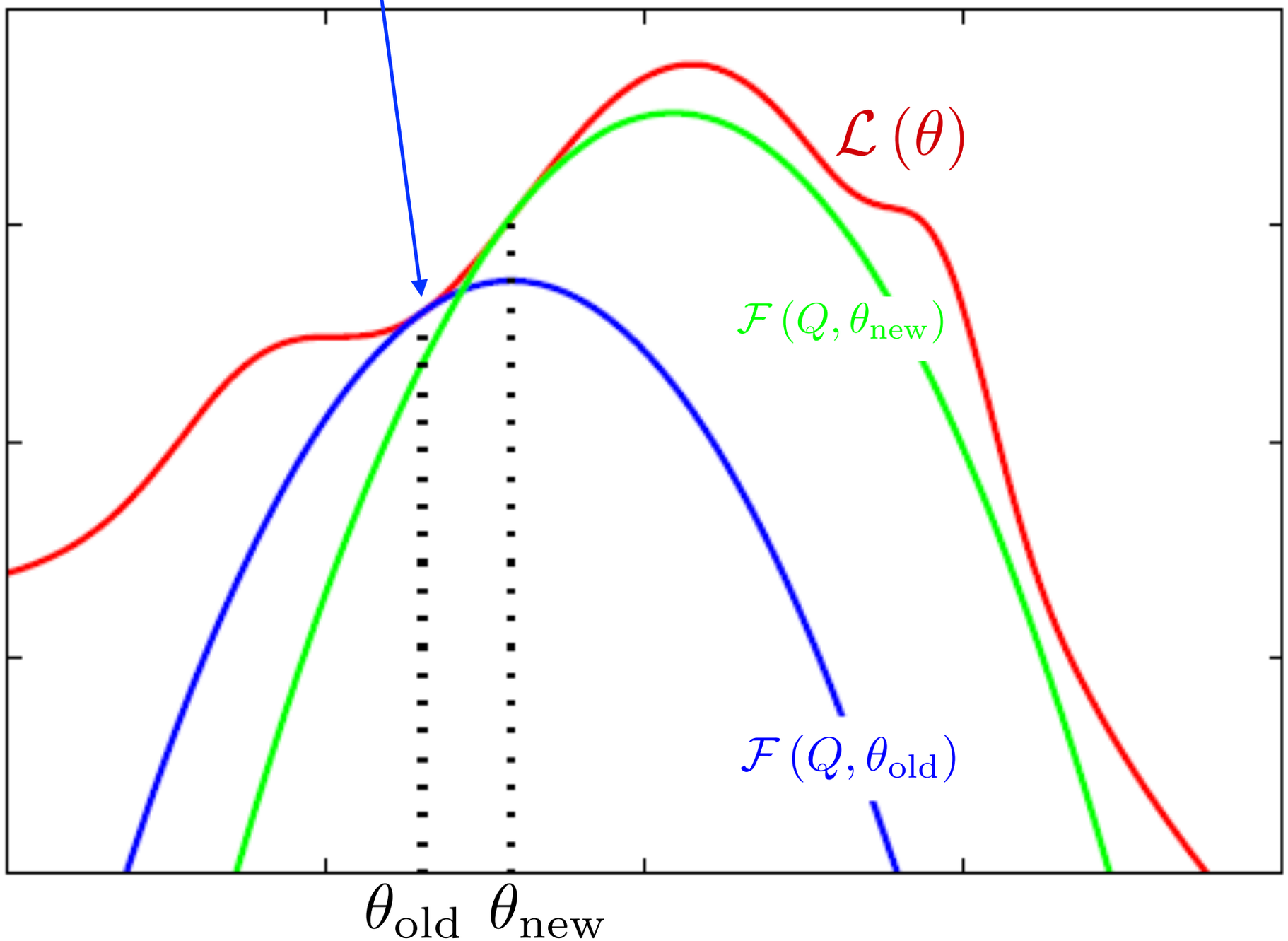
$$\textbf{M step:} \qquad \theta_{k+1} \leftarrow \underset{\theta}{\arg\max} \;\; \mathcal{F}(Q_{k+1}, \theta)$$

It is guaranteed that the likelihood will **never decrease** during this procedure.

General philosophy:
use smoothing estimates to guesstimate the state of the
latent variables, given current model parameters. Then use this fictitious
complete data to find new model parameters.

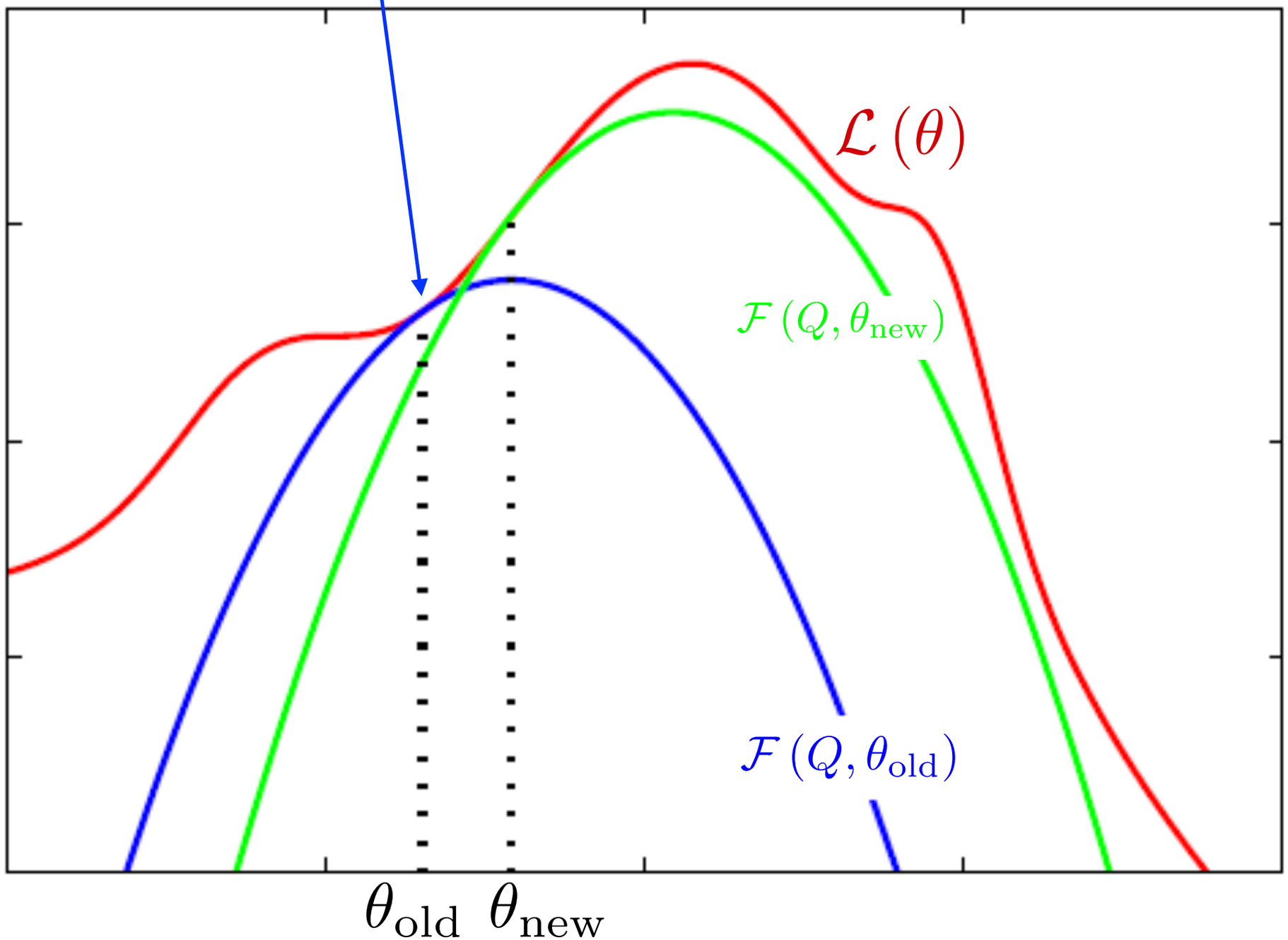

$\mathcal{L}(\theta)$
$\mathcal{F}(Q, \theta_{\text{new}})$
$\mathcal{F}(Q, \theta_{\text{old}})$
θ_{old}
θ_{new}

E-step: $Q_{k+1}(\mathbf{z}) = \mathrm{P}(\mathbf{z}|\mathbf{x}, \theta_k)$



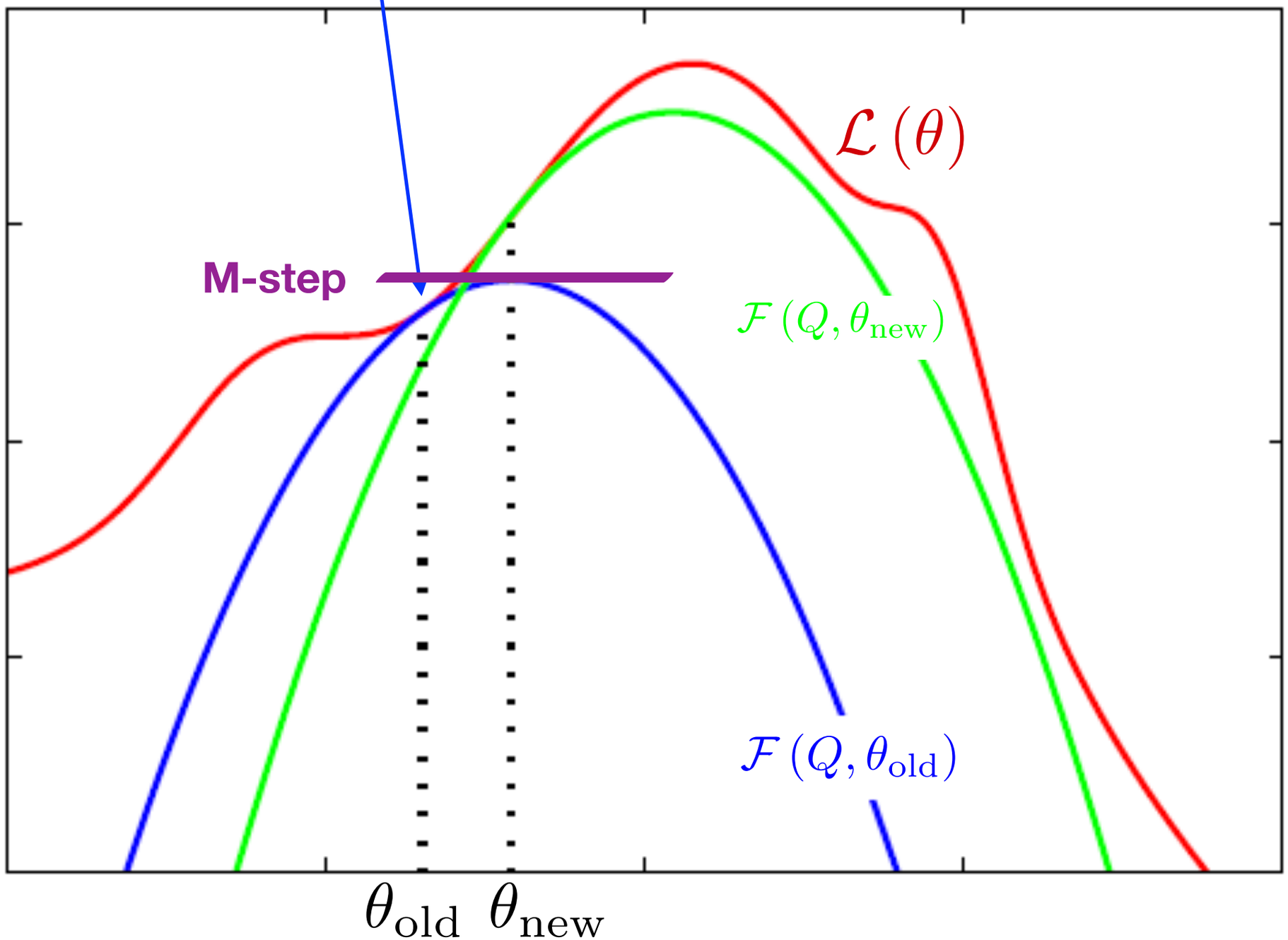
E-step: $Q_{k+1}(\mathbf{z}) = \mathrm{P}(\mathbf{z}|\mathbf{x}, \theta_k)$

The approximation is locally exact



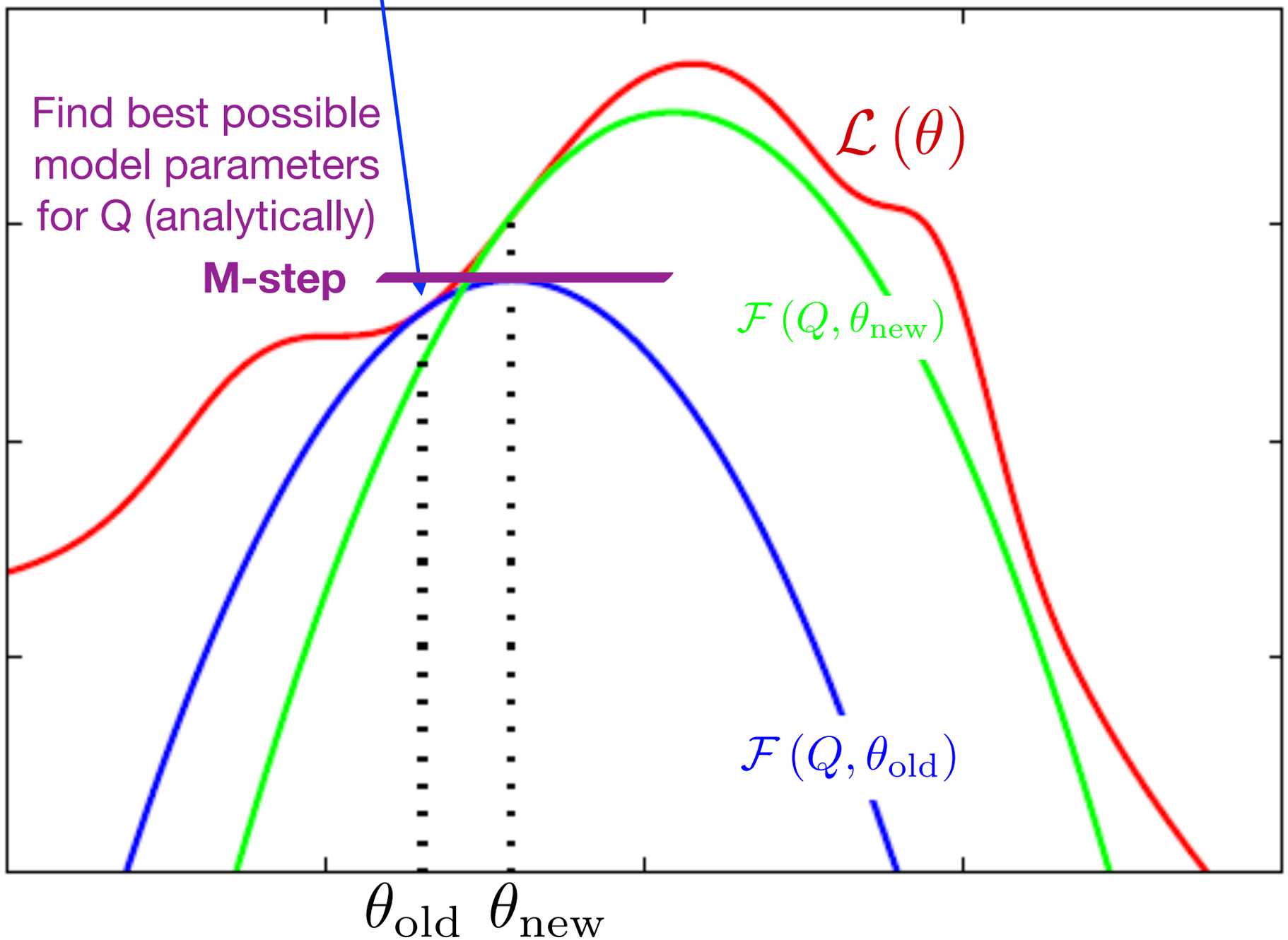
E-step: $Q_{k+1}(\mathbf{z}) = \mathrm{P}(\mathbf{z}|\mathbf{x}, \theta_k)$

The approximation is locally exact



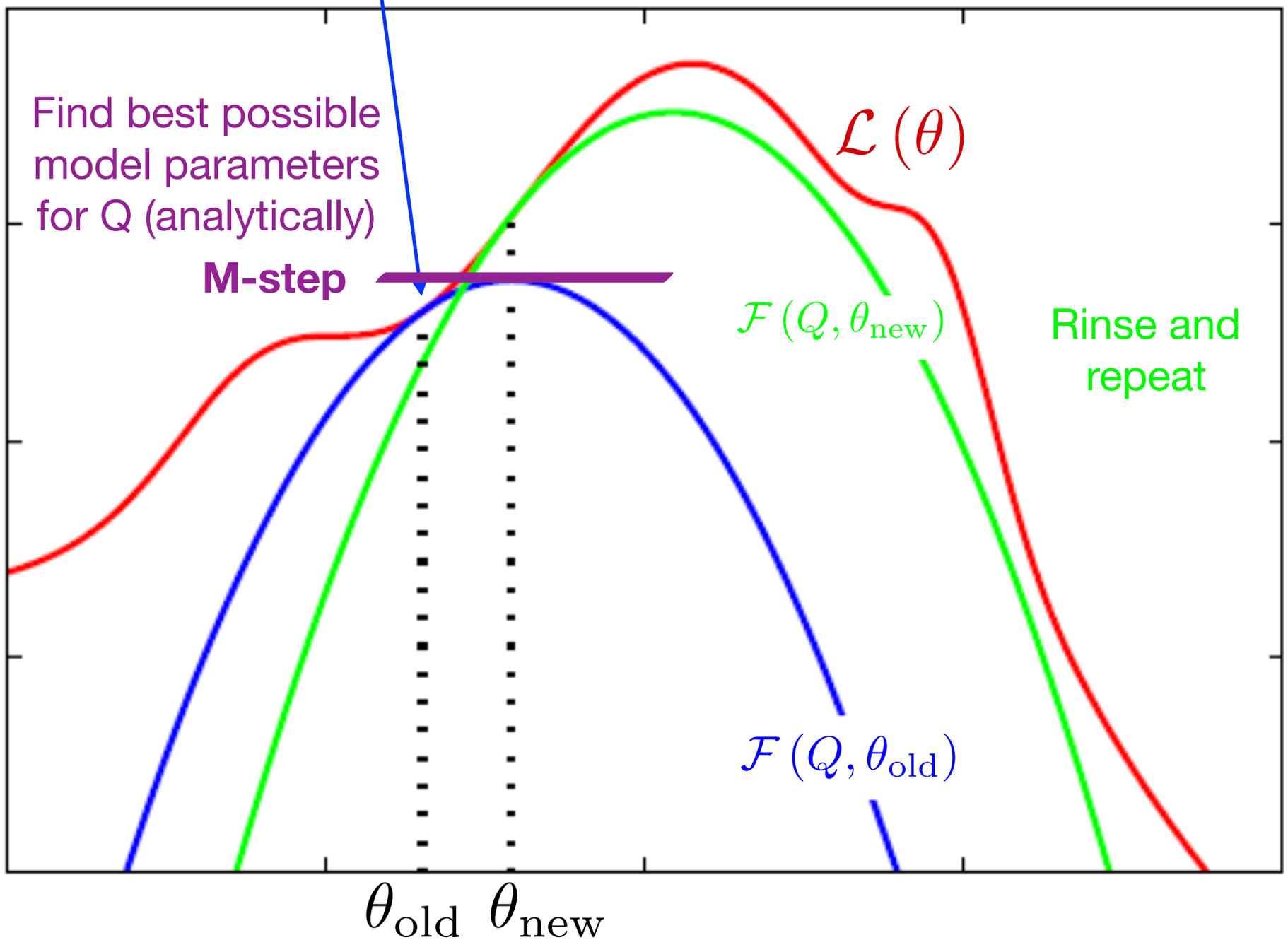
E-step: $Q_{k+1}(\mathbf{z}) = \mathrm{P}(\mathbf{z}|\mathbf{x}, \theta_k)$

The approximation is locally exact



E-step: $Q_{k+1}(\mathbf{z}) = \mathrm{P}(\mathbf{z}|\mathbf{x}, \theta_k)$

The approximation is locally exact



Lab: Artie is away
We'll do some basic Bayesian review and some slower derivations

Next: EM for LDS, sampling