CS 287: Advanced Robotics
Fall 2009

Lecture 21:
HMMs, Bayes filter, smoother, Kalman filters

Pieter Abbeel
UC Berkeley EECS

Overview

- Thus far:
  - Optimal control and reinforcement learning
  - We always assumed we got to observe the state at each time and the challenge was to choose a good action
- Current and next set of lectures
  - *The state is not observed*
  - Instead, we get some sensory information about the state

  → Challenge: compute a probability distribution over the state which accounts for the sensory information ("evidence") which we have observed.

Examples

- **Helicopter**
  - A choice of state: position, orientation, velocity, angular rate
  - Sensors:
    - GPS : noisy estimate of position (sometimes also velocity)
    - Inertial sensing unit: noisy measurements from (i) 3-axis gyro [=angular rate sensor], (ii) 3-axis accelerometer [=measures acceleration + gravity; e.g., measures (0,0,0) in free-fall], (iii) 3-axis magnetometer
- **Mobile robot inside building**
  - A choice of state: position and heading
  - Sensors:
    - Odometry (=sensing motion of actuators): e.g., wheel encoders
    - Laser range finder: measures time of flight of a laser beam between departure and return (return is typically happening when hitting a surface that reflects the beam back to where it came from)

Probability review

For any random variables X, Y we have:

- **Definition of conditional probability:**

  $P(X=x \mid Y=y) = P(X=x, Y=y) / P(Y=y)$

- **Chain rule:** (follows directly from the above)

  $P(X=x, Y=y) = P(X=x)\, P(Y=y \mid X=x)$

  $= P(Y=y)\, P(X=x \mid Y=y)$

- **Bayes rule:** (really just a re-ordering of terms in the above)

  $P(X=x \mid Y=y) = P(Y=y \mid X=x)\, P(X=x) / P(Y=y)$

- **Marginalization:**

  $P(X=x) = \sum_y P(X=x, Y=y)$

Note: no assumptions beyond X, Y being random variables are made for any of these to hold true (and when we divide by something, that something is not zero)

Probability review

For any random variables X, Y, Z, W we have:

- **Conditional probability:** (can condition on a third variable z throughout)

  P(X=x | Y=y, Z=z) = P(X=x, Y=y | Z=z) / P(Y=y | Z=z)

- **Chain rule:**

  P(X=x, Y=y, Z=z, W=w) = P(X=x) P(Y=y | X=x) P(Z=z | X=x, Y=y) P(W=w| X=x, Y=y, Z=z)

- **Bayes rule:** (can condition on other variable z throughout)

  P(X=x | Y=y, Z=z) = P(Y=y | X=x, Z=z) P(X=x | Z=z) / P(Y=y | Z=z)

- **Marginalization:**

  P(X=x | W=w) = $\sum_{y,z}$ P(X=x, Y=y, Z=z | W=w)

Note: no assumptions beyond X, Y, Z, W being random variables are made for any of these to hold true (and when we divide by something, that something is not zero)

Independence

- Two random variables X and Y are independent iff

  for all x, y : P(X=x, Y=y) = P(X=x) P(Y=y)

- Representing a probability distribution over a set of random variables X_1, X_2, …, X_T in its most general form can be expensive.
  - E.g., if all X_i are binary valued, then there would be a total of 2^T possible instantiations and it would require 2^T-1 numbers to represent the probability distribution.
- However, if we assumed the random variables were independent, then we could very compactly represent the joint distribution as follows:
  - $P(X_1=x_1, X_2=x_2, \ldots, X_T=x_T) = P(X_1=x_1) P(X_2=x_2) \ldots P(X_T=x_T)$
  - Thanks to the independence assumptions, for the binary case, we went from requiring 2^T-1 parameters, to only requiring T parameters!
- Unfortunately independence is often too strong an assumption …

Conditional independence

- Two random variables X and Y are conditionally independent given a third random variable Z iff

  for all x, y, z : $P(X=x, Y=y \mid Z=z) = P(X=x \mid Z=z)\, P(Y=y \mid Z=z)$

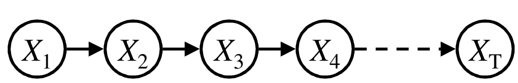
- Chain rule (which holds true for all distributions, no assumptions needed):
  - $P(X=x,Y=y,Z=z,W=w) = P(X=x)P(Y=y|X=x)P(Z=z|X=x,Y=y)P(W=w|X=x,Y=y,Z=z)$

  → For binary variables the representation requires $1 + 2*1 + 4*1 + 8*1 = 2^4-1$ numbers (just like a full joint probability table)

- Now assume Z independent of X given Y, and assume W independent of X and Y given Z, then we obtain:
  - $P(X=x,Y=y,Z=z,W=w) = P(X=x)P(Y=y|X=x)P(Z=z|Y=y)P(W=w|Z=z)$
  - → For binary variables the representation requires $1 + 2*1 + 2*1 + 2*1 = 1+(4-1)*2$ numbers --- significantly less!!

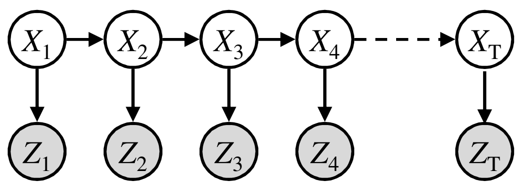
Markov Models

- Models a distribution over a set of random variables $X_1, X_2, \ldots, X_T$ where the index is typically associated with some notion of time.
- Markov models make the assumption:
  - X_t is independent of $X_1, \ldots, X_{t-2}$ when given X_{t-1}
- Chain rule: (always holds true, not just in Markov models!)
  - $P(X_1 = x_1, X_2 = x_2, \ldots, X_T = x_T) = \prod_t P(X_t = x_t \mid X_{t-1} = x_{t-1}, X_{t-2} = x_{t-2}, \ldots, X_1 = x_1)$
- Now apply the Markov conditional independence assumption:
  - $P(X_1 = x_1, X_2 = x_2, \ldots, X_T = x_T) = \prod_t P(X_t = x_t \mid X_{t-1} = x_{t-1})$ (1)
  - → in binary case: $1 + 2*(T-1)$ numbers required to represent the joint distribution over all variables (vs. $2^T - 1$)
- Graphical representation: a variable X_t receives an arrow from the variables appearing in its conditional probability in the expression for the joint distribution (1) [called a Bayesian network or Bayes net representation]

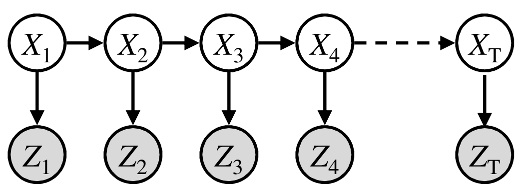


Hidden Markov Models

- Underlying Markov model over states X_t
  - Assumption 1: X_t independent of $X_1, \ldots, X_{t-2}$ given X_{t-1}
- For each state X_t there is a random variable Z_t which is a sensory measurement of X_t
  - Assumption 2: Z_t is assumed conditionally independent of the other variables *given* X_t
- This gives the following graphical (Bayes net) representation:



Hidden Markov Models



$P(X_1=x_1, Z_1=z_1, X_2=x_2, Z_2=z_2, \ldots, X_T=x_T, Z_T=z_T) =$

- Chain rule: (no assumptions)

  $P(X_1 = x_1)$

  $P(Z_1 = z_1 \mid X_1 = x_1)$

  $P(X_2 = x_2 \mid X_1 = x_1, Z_1 = z_1)$

  $P(Z_2 = z_2 \mid X_1 = x_1, Z_1 = z_1, X_2 = x_2)$

  …

  $P(X_T = x_T \mid X_1 = x_1, Z_1 = z_1, \ldots, X_{T-1} = x_{T-1}, Z_{T-1} = z_{T-1})$

  $P(Z_T = z_T \mid X_1 = x_1, Z_1 = z_1, \ldots, X_{T-1} = x_{T-1}, Z_{T-1} = z_{T-1}, X_T = x_T)$

- HMM assumptions:

  $P(X_1 = x_1)$

  $P(Z_1 = z_1 \mid X_1 = x_1)$

  $P(X_2 = x_2 \mid X_1 = x_1)$

  $P(Z_2 = z_2 \mid X_2 = x_2)$

  …

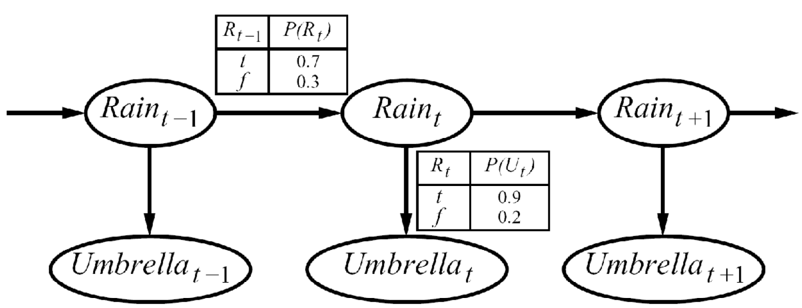
  $P(X_T = x_T \mid X_{T-1} = x_{T-1})$

  $P(Z_T = z_T \mid X_T = x_T)$

Mini quiz

- What would the graph look like for a Bayesian network with no conditional independence assumptions?
- Our particular choice of ordering of variables in the chain rule enabled us to easily incorporate the HMM assumptions. What if we had chosen a different ordering in the chain rule expansion?

Example



- The HMM is defined by:
  - Initial distribution: $P(X_1)$
  - Transitions: $P(X|X_{-1})$
  - Observations: $P(Z|X)$

Real HMM Examples

- Robot localization:
  - Observations are range readings (continuous)
  - States are positions on a map (continuous)
- Speech recognition HMMs:
  - Observations are acoustic signals (continuous valued)
  - States are specific positions in specific words (so, tens of thousands)
- Machine translation HMMs:
  - Observations are words (tens of thousands)
  - States are translation options

Filtering / Monitoring

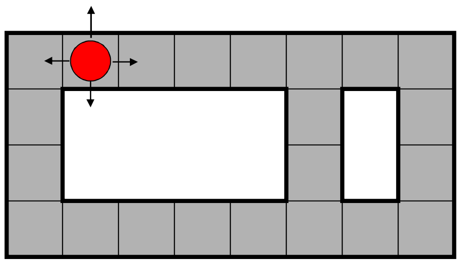
- Filtering, or monitoring, is the task of tracking the distribution

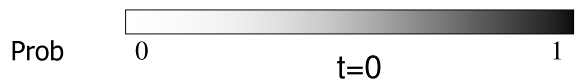
  $P(X_t \mid Z_1 = z_1 , Z_2 = z_2 , \ldots, Z_t = z_t)$

  over time. This distribution is called the belief state.
- We start with $P(X_0)$ in an initial setting, usually uniform
- As time passes, or we get observations, we update the belief state.
- The Kalman filter was invented in the 60's and first implemented as a method of trajectory estimation for the Apollo program. [See course website for a historical account on the Kalman filter. "From Gauss to Kalman"]

Example: Robot Localization

Example from Michael Pfeiffer





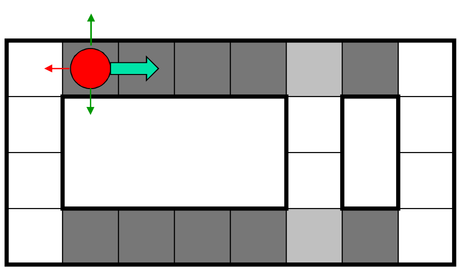
t=0

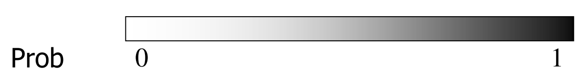
Sensor model: never more than 1 mistake

Know the heading (North, East, South or West)

Motion model: may not execute action with small prob.

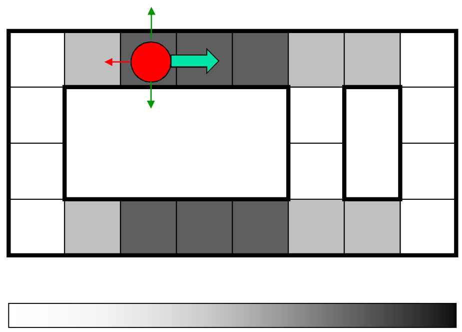
Example: Robot Localization





t=1

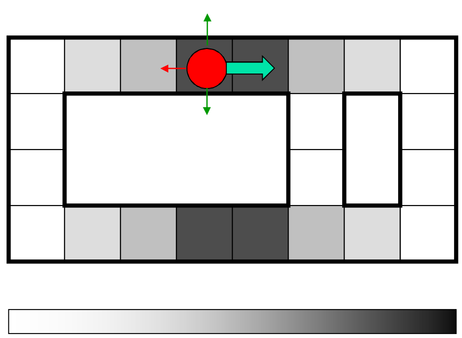
Example: Robot Localization



Prob 0 1

t=2

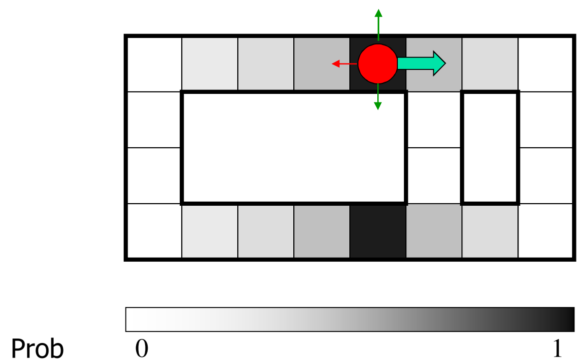
Example: Robot Localization



Prob 0 1

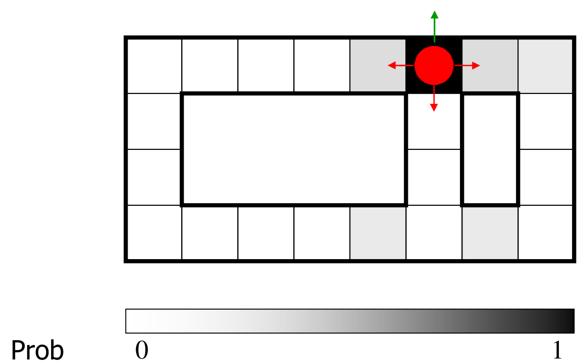
t=3

Example: Robot Localization



t=4

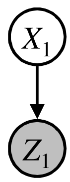
Example: Robot Localization



t=5

Inference: Base Cases

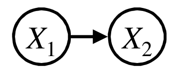
Incorporate observation



$$P(X_1|z_1)$$

$$\begin{aligned} P(x_1|z_1) &= P(x_1, z_1)/P(z_1) \\ &\propto P(x_1, z_1) \\ &= P(x_1)P(z_1|x_1) \end{aligned}$$

Time update



$$P(X_2)$$

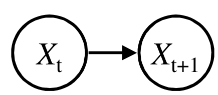
$$\begin{aligned} P(x_2) &= \sum_{x_1} P(x_1, x_2) \\ &= \sum_{x_1} P(x_1)P(x_2|x_1) \end{aligned}$$

Time update

- Assume we have current belief P(X | evidence to date)

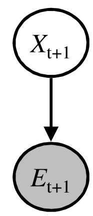
$$P(x_t|e_{1:t})$$

- Then, after one time step passes:



$$P(x_{t+1}|e_{1:t}) = \sum_{x_t} P(x_{t+1}|x_t)P(x_t|e_{1:t})$$

Observation update



- Assume we have:

$$P(x_{t+1}|e_{1:t})$$

- Then:

$$P(x_{t+1}|e_{1:t+1}) \propto P(e_{t+1}|x_{t+1})P(x_{t+1}|e_{1:t})$$

Algorithm

- Init $P(x_1)$ [e.g., uniformly]
- Observation update for time 0:

$$P(x_1|z_1) \propto P(z_1|x_1)P(x_1)$$

- For t = 1, 2, …
  - Time update

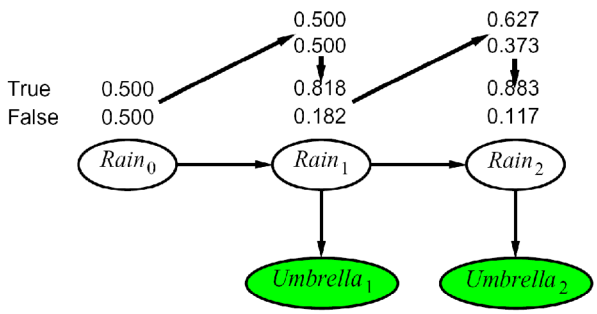
$$P(x_{t+1}|z_{1:t}) = \sum_{x_t} P(x_{t+1}|x_t)P(x_t|z_{1:t})$$

  - Observation update

$$P(x_{t+1}|z_{1:t+1}) \propto P(z_{t+1}|x_{t+1})P(x_{t+1}|z_{1:t})$$

- For continuous state / observation spaces: simply replace summation by integral

Example HMM



R_{t-1}	$P(R_t)$
T	0.7
F	0.3

R_t	$P(U_t)$
T	0.9
F	0.2

The Forward Algorithm

- Time/dynamics update and observation update in one:

$$\begin{aligned} P(x_t, z_{1:t}) &= \sum_{x_{t-1}} P(x_{t-1}, x_t, z_{1:t}) \\ &= \sum_{x_{t-1}} P(x_{t-1}, z_{1:t-1}) P(x_t|x_{t-1}) P(z_t|x_t) \\ &= P(z_t|x_t) \sum_{x_{t-1}} P(x_t|x_{t-1}) P(x_{t-1}, z_{1:t-1}) \end{aligned}$$

- → recursive update
- Normalization:
  - Can be helpful for numerical reasons
  - However: lose information!

  → Can renormalize (for numerical reasons) + keep track of the normalization factor (to enable recovering all information)

The likelihood of the observations

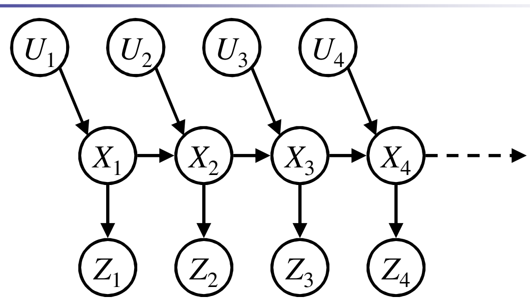
$$P(z_{1:t}) = \sum_{x_1, x_2, \ldots, x_t} P(x_{1:t}, z_{1:t}) = \sum_{x_1, x_2, \ldots, x_t} \prod_{k=1}^{t-1} P(x_{k+1}|x_k) P(z_k|x_k) P(z_t|x_t)$$

- The forward algorithm first sums over x_1, then over x_2 and so forth, which allows it to efficiently compute the likelihood at all times t, indeed:

$$P(z_{1:t}) = \sum_{x_t} P(x_t, z_{1:t})$$

- Relevance:
  - Compare the fit of several HMM models to the data
  - Could optimize the dynamics model and observation model to maximize the likelihood
  - Run multiple simultaneous trackers --- retain the best and split again whenever applicable (e.g., loop closures in SLAM, or different flight maneuvers)

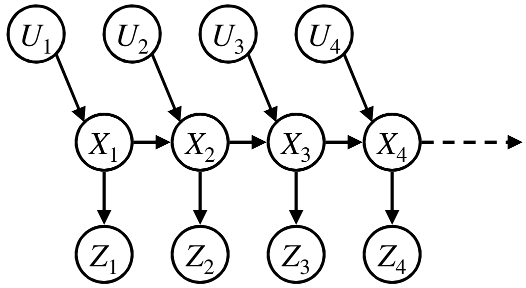
With control inputs



We know track:

$$P(x_t|z_{1:t}, u_{1:t})$$

With control inputs



- Control inputs known:
  - They can be simply seen as selecting a particular dynamics function
- Control inputs unknown:
  - Assume a distribution over them
- Above drawing assumes open-loop controls. This is rarely the case in practice. [Markov assumption is rarely the case either. Both assumptions seem to have given quite satisfactory results.]

Smoothing

- Thus far, filtering, which finds:
  - The distribution over states at time t given all evidence until time t:

$$P(x_t|z_{1:t}, u_{1:t})$$

  - The likelihood of the evidence up to time t:

$$P(z_{1:t}|u_{1:t})$$

- How about?

$$P(x_t|z_{1:T}, u_{1:T})$$

  - T < t : can simply run the forward algorithm until time t, but stop incorporating evidence from time T+1 onwards
  - T > t : need something else

Smoothing

$$
\begin{aligned}
P(x_t|z_{1:T}) &\propto P(x_t, z_{1:T}) \\
&= \sum_{x_1, x_2, \ldots, x_{t-1}, x_{t+1}, \ldots, x_T} P(x_{1:T}, z_{1:T}) \\
&= \sum_{x_1, x_2, \ldots, x_{t-1}, x_{t+1}, \ldots, x_T} \prod_{k=1}^{T} P(x_k|x_{k-1}) P(z_k|x_k)
\end{aligned}
$$

- Sum as written has a number of terms exponential in T
- Key idea: order in which variables are being summed out affects computational complexity
  - *Forward algorithm* exploits summing out $x_1, x_2, \ldots, x_{t-1}$ in this order
  - Can similarly run a *backward algorithm*, which sums out $x_T, x_{T-1}, \ldots, x_{t+1}$ in this order

Smoothing

$$
\begin{aligned}
P(x_t, z_{1:T}) &= \sum_{x_1, x_2, \ldots, x_{t-1}, x_{t+1}, \ldots, x_T} P(x_{1:T}, z_{1:T}) \\
&= \sum_{x_1, x_2, \ldots, x_{t-1}, x_{t+1}, \ldots, x_T} \prod_{k=1}^{T} P(x_k|x_{k-1}) P(z_k|x_k) \\
&= \left(\sum_{x_1, x_2, \ldots, x_{t-1}} \prod_{k=1}^{t} P(x_k|x_{k-1}) P(z_k|x_k) \right) \left(\sum_{x_{t+1}, x_{t+2}, \ldots, x_T} \prod_{k=t+1}^{T} P(x_k|x_{k-1}) P(z_k|x_k) \right)
\end{aligned}
$$

Forward algorithm computes this — Backward algorithm computes this

- Can be easily verified from the equations:
  - The factors in the right parentheses only contain $X_{t+1}, \ldots, X_T$, hence they act as a constant when summing out over $X_1, \ldots, X_{t-1}$ and can be brought outside the summation
- Can also be read off from the Bayes net graph / conditional independence assumptions:
  - $X_1, \ldots, X_{t-1}$ are conditionally of $X_{t+1}, \ldots, X_T$ *given* X_t

Backward algorithm

- Sum out x_T:

$$\begin{aligned}
P(x_t, e_{1:T}) &= \sum_{x_1, x_2, \ldots, x_{t-1}, x_{t+1}, \ldots, x_T} \prod_{k=1}^{T} P(x_k|x_{k-1}) P(e_k|x_k) \\
&= \sum_{x_1, x_2, \ldots, x_{t-1}, x_{t+1}, \ldots, x_{T-1}} \sum_{x_T} \prod_{k=1}^{T} P(x_k|x_{k-1}) P(e_k|x_k) \\
&= \sum_{x_1, x_2, \ldots, x_{t-1}, x_{t+1}, \ldots, x_{T-1}} \prod_{k=1}^{T-1} P(x_k|x_{k-1}) P(e_k|x_k) \sum_{x_T} P(x_T|x_{T-1}) P(e_T|x_T) \\
&= \sum_{x_1, x_2, \ldots, x_{t-1}, x_{t+1}, \ldots, x_{T-1}} \prod_{k=1}^{T-1} P(x_k|x_{k-1}) P(e_k|x_k) f_{T-1}(x_{T-1})
\end{aligned}$$

- Can recursively compute for l=T, T-1, …:

$$f_{l-1}(x_{l-1}) = \sum_{x_l} P(x_l|x_{l-1}) P(e_l|x_l) f_l(x_l)$$

Smoother algorithm

- Run forward algorithm, which gives
  - $P(x_t, z_1, \ldots, z_t)$ for all t
- Run backward algorithm, which gives
  - $f_t(x_t)$ for all t
- Find
  - $P(x_t, z_1, \ldots, z_T) = P(x_t, z_1, \ldots, z_t)\, f_t(x_t)$
  - If desirable, can renormalize and find $P(x_t \mid z_1, \ldots, z_T)$

Bayes filters

- Recursively compute
  - $P(x_t , z_{1:t-1}) = \sum_{x_{t-1}} P(x_t \mid x_{t-1}) P(x_{t-1} \mid z_{1:t-1})$
  - $P(x_t , z_{1:t}) = P(x_t, z_{1:t-1}) P(z_t \mid x_t)$
- Tractable cases:
  - State space finite and sufficiently small

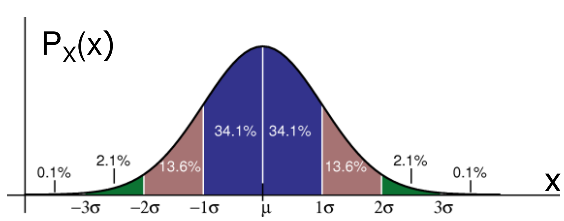
    (what we have in some sense considered so far)
  - Systems with linear dynamics and linear observations and Gaussian noise

    → Kalman filtering

Univariate Gaussian

- Gaussian distribution with mean μ, and standard deviation σ:

$$N(\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$



Properties of Gaussians

$$N(\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{(x-\mu)^2}{2\sigma^2})$$

- Mean:

$$EX = \int x \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{(x-\mu)^2}{2\sigma^2}) = \mu$$

- Variance:

$$E((X-\mu)^2) = \int (x-\mu)^2 \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{(x-\mu)^2}{2\sigma^2}) = \sigma^2$$

Central limit theorem (CLT)

- Classical CLT:
  - Let X_1, X_2, … be an infinite sequence of *independent* random variables with E $X_i = \mu$, $E(X_i - \mu)^2 = \sigma^2$
  - Define $Z_n = ((X_1 + \ldots + X_n) - n\,\mu) / (\sigma\, n^{1/2})$
  - Then for the limit of n going to infinity we have that Z_n is distributed according to N(0,1)
- Crude statement: things that are the result of the addition of lots of small effects tend to become Gaussian.

Multi-variate Gaussians

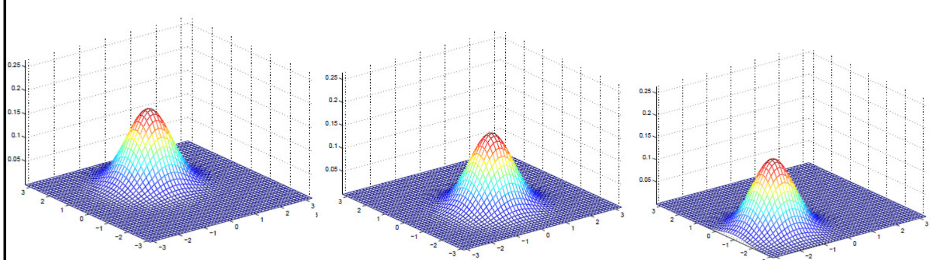
$$p(x;\mu,\Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right)$$

$$EX = \int x p(x;\mu,\Sigma) = \mu$$

$$E[(X_i-\mu_i)(X_j-\mu_j)] = \int (x_i-\mu_i)(x_j-\mu_j)p(x;\mu,\Sigma) = \Sigma_{ij}$$

$$\int \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right) = 1$$

Multi-variate Gaussians: examples

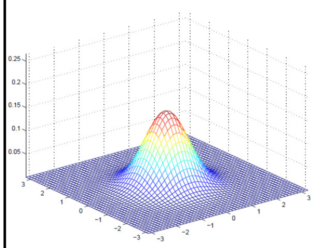


- μ = [1; 0]
- Σ = [1 0; 0 1]

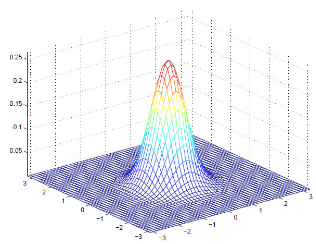
- μ = [-.5; 0]
- Σ = [1 0; 0 1]

- μ = [-1; -1.5]
- Σ = [1 0; 0 1]

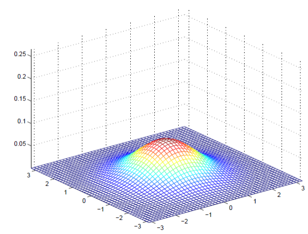
Multi-variate Gaussians: examples



- μ = [0; 0]
- Σ = [1 0 ; 0 1]

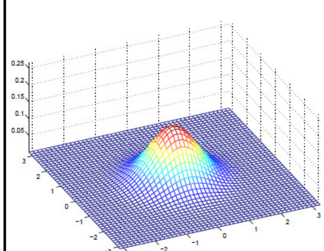


- μ = [0; 0]
- Σ = [.6 0 ; 0 .6]

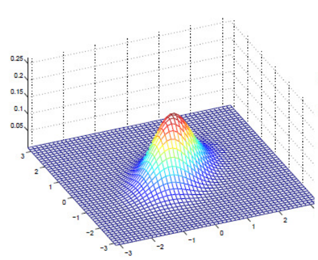


- μ = [0; 0]
- Σ = [2 0 ; 0 2]

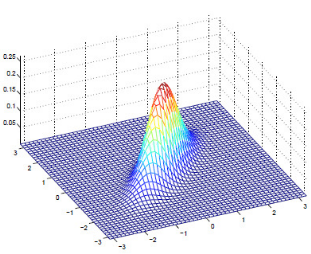
Multi-variate Gaussians: examples



- μ = [0; 0]
- Σ = [1 0; 0 1]

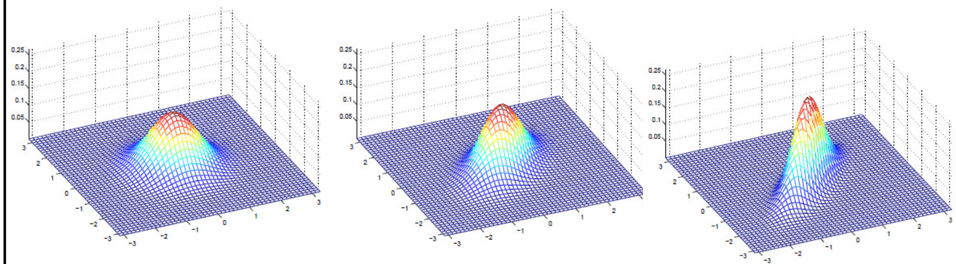


- μ = [0; 0]
- Σ = [1 0.5; 0.5 1]



- μ = [0; 0]
- Σ = [1 0.8; 0.8 1]

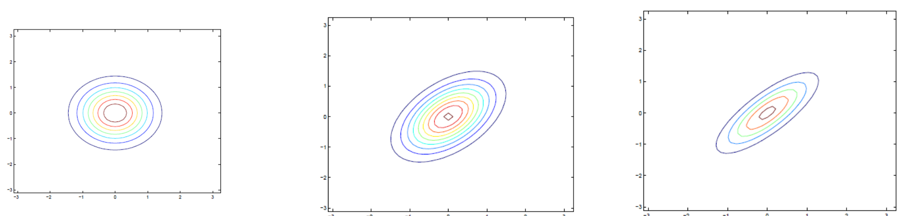
Multi-variate Gaussians: examples



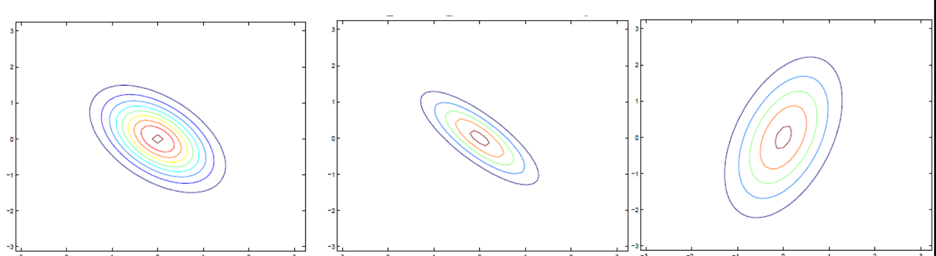
- $\mu = [0; 0]$
- $\Sigma = [1\ \ 0; 0\ \ 1]$

- $\mu = [0; 0]$
- $\Sigma = [1\ \ 0.5; 0.5\ \ 1]$

- $\mu = [0; 0]$
- $\Sigma = [1\ \ 0.8; 0.8\ \ 1]$



Multi-variate Gaussians: examples



- $\mu = [0; 0]$
- $\Sigma = [1\ \ -0.5\ ; -0.5\ \ 1]$

- $\mu = [0; 0]$
- $\Sigma = [1\ \ -0.8\ ; -0.8\ \ 1]$

- $\mu = [0; 0]$
- $\Sigma = [3\ \ 0.8\ ; 0.8\ \ 1]$

Discrete Kalman Filter

Estimates the state x of a discrete-time controlled process that is governed by the linear stochastic difference equation

$$x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t$$

with a measurement

$$z_t = C_t x_t + \delta_t$$

47

Components of a Kalman Filter

A_t Matrix (nxn) that describes how the state evolves from t to $t-1$ without controls or noise.

B_t Matrix (nxl) that describes how the control u_t changes the state from t to $t-1$.

C_t Matrix (kxn) that describes how to map the state x_t to an observation z_t.

ε_t, δ_t Random variables representing the process and measurement noise that are assumed to be independent and normally distributed with covariance R_t and Q_t respectively.

48

Linear Gaussian Systems: Initialization

- Initial belief is normally distributed:

$$bel(x_0) = N\left(x_0; \mu_0, \Sigma_0\right)$$

53

Linear Gaussian Systems: Dynamics

- Dynamics are linear function of state and control plus additive noise:

$$x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t$$

$$p(x_t \mid u_t, x_{t-1}) = N\left(x_t; A_t x_{t-1} + B_t u_t, R_t\right)$$

$$\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1}) \qquad bel(x_{t-1})\, dx_{t-1}$$

$$\Downarrow \qquad\qquad \Downarrow$$

$$\sim N\left(x_t; A_t x_{t-1} + B_t u_t, R_t\right) \quad \sim N\left(x_{t-1}; \mu_{t-1}, \Sigma_{t-1}\right)$$

54

Linear Gaussian Systems: Dynamics

$$\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1}) \qquad bel(x_{t-1})\, dx_{t-1}$$

$$\Downarrow \qquad\qquad \Downarrow$$

$$\sim N\left(x_t; A_t x_{t-1} + B_t u_t, R_t\right) \quad \sim N\left(x_{t-1}; \mu_{t-1}, \Sigma_{t-1}\right)$$

$$\Downarrow$$

$$\overline{bel}(x_t) = \eta \int \exp\left\{-\frac{1}{2}(x_t - A_t x_{t-1} - B_t u_t)^T R_t^{-1} (x_t - A_t x_{t-1} - B_t u_t)\right\}$$

$$\exp\left\{-\frac{1}{2}(x_{t-1} - \mu_{t-1})^T \Sigma_{t-1}^{-1} (x_{t-1} - \mu_{t-1})\right\} dx_{t-1}$$

$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = A_t \mu_{t-1} + B_t u_t \\ \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t \end{cases}$$

55

Proof: completion of squares

- To integrate out x_{t-1}, re-write the integrand in the format:

$$\int f(\mu_{t-1}, \Sigma_{t-1}, x_t) \left(\frac{1}{(2\pi)^{\frac{d}{2}} |S|^{\frac{1}{2}}} \exp\left(\frac{-1}{2}(x_{t-1} - m) S^{-1} (x_{t-1} - m) \right) \right) dx_{t-1}$$

- This integral is readily computed (integral of a multivariate Gaussian times a constant = that constant) to be

$$f(\mu_{t-1}, \Sigma_{t-1}, x_t)$$

- Inspection of f will show that it is a multi-variate Gaussian in x_t with the mean and covariance as shown on previous slide.

Properties of Gaussians

- We just showed:

$$\left.\begin{array}{r} X \sim N(\mu, \Sigma) \\ Y = AX + B \end{array}\right\} \quad \Rightarrow \quad Y \sim N(A\mu + B, A\Sigma A^T)$$

- We stay in the "Gaussian world" as long as we start with Gaussians and perform only linear transformations.
- Now we know this, we could find μ_Y and Σ_Y without computing integrals by directly computing the expected values:

$$\mathrm{E}[Y] = \mathrm{E}[AX + B] = A\mathrm{E}[X] + B = A\mu + B$$

$$\begin{aligned} \Sigma_{YY} &= \mathrm{E}[(Y - \mathrm{E}[Y])(Y - \mathrm{E}[Y])^\top] = \mathrm{E}[(AX + B - A\mu - B)(AX + B - A\mu - B)^\top] \\ &= \mathrm{E}[A(X - \mu)(X - \mu)^\top A^\top] = A\mathrm{E}[(X - \mu)(X - \mu)^\top]A^\top = A\Sigma A^\top \end{aligned}$$

Self-quiz

Test your understanding of the completion of squares trick! Let $A \in \mathbf{R}^{n\times n}$ be a positive definite matrix, $b \in \mathbf{R}^n$, and $c \in \mathbf{R}$. Prove that

$$\int_{x\in\mathbf{R}^n} \exp\left(-\frac{1}{2}x^T Ax - x^T b - c\right) dx = \frac{(2\pi)^{n/2}}{|A|^{1/2}\exp(c - \frac{1}{2}b^T A^{-1} b)}.$$

Linear Gaussian Systems: Observations

- Observations are linear function of state plus additive noise:

$$z_t = C_t x_t + \delta_t$$

$$p(z_t \mid x_t) = N(z_t; C_t x_t, Q_t)$$

$$bel(x_t) = \eta \quad p(z_t \mid x_t) \qquad \overline{bel}(x_t)$$

$$\Downarrow \qquad\qquad \Downarrow$$

$$\sim N(z_t; C_t x_t, Q_t) \qquad \sim N(x_t; \bar{\mu}_t, \bar{\Sigma}_t)$$

59

Linear Gaussian Systems: Observations

$$bel(x_t) = \eta \quad p(z_t \mid x_t) \qquad \overline{bel}(x_t)$$

$$\Downarrow \qquad\qquad \Downarrow$$

$$\sim N(z_t; C_t x_t, Q_t) \qquad \sim N(x_t; \bar{\mu}_t, \bar{\Sigma}_t)$$

$$\Downarrow$$

$$bel(x_t) = \eta \exp\left\{-\frac{1}{2}(z_t - C_t x_t)^T Q_t^{-1}(z_t - C_t x_t)\right\} \exp\left\{-\frac{1}{2}(x_t - \bar{\mu}_t)^T \bar{\Sigma}_t^{-1}(x_t - \bar{\mu}_t)\right\}$$

$$bel(x_t) = \begin{cases} \mu_t = \bar{\mu}_t + K_t(z_t - C_t \bar{\mu}_t) \\ \Sigma_t = (I - K_t C_t)\bar{\Sigma}_t \end{cases} \quad \text{with} \quad K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$$

60

Proof: completion of squares

- Re-write the expression for *bel(x_t)* in the format:

$$f(\bar{\mu}_t, \bar{\Sigma}_t, C_t, Q_t)\left(\frac{1}{(2\pi)^{\frac{d}{2}}|S|^{\frac{1}{2}}}\exp\left(\frac{-1}{2}(x_t - m)S^{-1}(x_t - m)\right)\right)$$

- f is the normalization factor
- The expression in parentheses is a multi-variate Gaussian in x_t. Its parameters m and S can be identified to satisfy the expressions for the mean and covariance on the previous slide.

Kalman Filter Algorithm

Algorithm **Kalman_filter**(μ_{t-1}, Σ_{t-1}, u_t, z_t):

Prediction:

$$\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$$
$$\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$$

Correction:

$$K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$$
$$\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$$
$$\Sigma_t = (I - K_t C_t)\bar{\Sigma}_t$$

Return μ_t, Σ_t

62

How to derive these updates

- Simply work through the integrals
  - Key "trick": completion of squares

- If your derivation results in a different format → apply matrix inversion lemma to prove equivalence

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}$$