Session 11

I. Announcements [5 minutes]

- **Homework 5A is due 11/14. Get partners and get busy as this is a 2 partner.**

Overview

My sections are a bit ahead of lecture this week, so it occurs to me that we're going too fast. As such we'll cover 2 topics today and then the floor is open for questions. The topics we'll be covering will hopefully give you an introduction to the material you'll need to know for Assignment 5 and I'm hoping your experience with Assignment 4 will fuel questions.

1. Dynamic Bayes Nets
    a. Constructing
    b. Exact Inference
    c. Approximate Inference
2. MCMC
    a. Why Monte Carlo?
    b. Why Markov Chains?
3. Project/Homework Questions and Noisy-OR

II. Dynamic Bayesian Networks

Dynamic Bayesian Networks (DBN) – a Bayesian network that represents a temporal probability model by having state variables $\mathbf{X_t}$ replicated over time slices with the same conditional independences. We also have evidence at each time slice $\mathbf{E_t}$. For simplicity we assume a 1st order Markov process → a node's parents are either in the current or previous time slice.

- DBNs take advantage of the sparseness of the temporal probability model, whereas the equivalent HMM assumes all internal state is dependent.
- DBNs can model arbitrary distributions (thus extending beyond the capabilities of a Kalman Filter) allowing it to capture nonlinearities other models cannot.
- *Constructing DBNs*
  - We need 3 broad types of information:
    1. a prior distribution on the initial variables: $\mathbf{P}(\mathbf{X_0})$
    2. a transition model: $\mathbf{P}(\mathbf{X_{t+1}} \mid \mathbf{X_t})$
    3. a sensor model: $\mathbf{P}(\mathbf{E_t} \mid \mathbf{X_t})$
  - In addition, we must specify a local and temporal topology of the nodes at the current state and the nodes at the previous state.
  - Since the transition & sensor models are assumed to be stationary they remain the same over time → *only need to specify for initial time slice.*
  - Issues we need to deal with:
    - *Noise*: we assume that our measurements are noisy, which we model with a **Gaussian error model**.
    - *Failure*: in the real-world, sensors fail – we need to model this effect.
      - *In order to properly handle sensor failure, the sensor model must explicitly include the possibility of failure.*
      - ***transient failure model*** – allocates a probability that the sensor will return some nonsense value. This has the effect of "*inertia*" to prevent radical shifts due to intermediate failures.
      - ***persistent failure model*** – describes how a sensor behaves under normal and failure conditions. In particular, we have a small probability of failure, but it also models the fact that sensors tend to remain broken.

- *Exact Inference* – given a sequence of *n* observations, we simply construct the necessary DBN of *n* time slices – a process known as ***unrolling***.
    - But naively constructing the unrolled network requires *O(t)* space and inference at each time step increases at *O(t)*.
    - A more efficient process uses ***variable elimination*** before proceeding to the next time slice – this is equivalent to starting at $\mathbf{X_t}$ with a new initial distribution determined by our variable elimination.
        - This process exactly mimics the operation of a recursive filtering update. This allows us to have constant space and time per slice.
        - *Unfortunately*, the constant is exponential in number of state variables.
        - *We cannot efficiently and exactly reason about the complex temporal processes represented by general DBNs.*
- *Approximate Inference* – to estimate inference on a DBN we need to overcome a few obstacles:
    - Overcoming these blocks relies on 2 observations.
        - Again, unrolling the network is inefficient. Again, we run the samples through the network one slice at a time. *We use the samples as approximate representations of the current state distribution*.
        - Generating the samples with naïve likelihood weighting will have ~0 probability of matching the evidence. Thus, w.h.p. the samples will be independent of the evidence and will have no weight.
            - Thus, we require exponential samples to get accuracy.
            - *Instead, we want to focus the set of samples on the high-probability regions of the sate space.* We simply throw out samples of very low weight.
    - **<u>particle filtering</u>** – leverages the above observations to make an efficient sampling algorithm that is ***consistent***. We begin with *N* samples from the prior distribution at time 0: $\mathbf{P}\left(\mathbf{X_0}\right)$. Then we use an <u>update cycle</u>:
        - Each sample is propagated to next time slice by sampling the next state value $\mathbf{x_{t+1}}$ given $\mathbf{x_t}$ using the transition model $\mathbf{P}\left(\mathbf{X_{t+1}} \mid \mathbf{X_t}\right)$.
        - Each sample is weighted by the likelihood it assigns to the new evidence: $\mathbf{P}\left(\mathbf{e_{t+1}} \mid \mathbf{x_{t+1}}\right)$ from the sensor model.
        - A new population of *N* samples is *resampled*: each new sample is selected proportional to its likelihood weight.

III. MCMC

- **Markov chain Monte Carlo (MCMC)** – a sampling technique that settles into a *dynamic equilibrium* such that the long-term fraction of time spent in each state is exactly its posterior probability given certain conditions.
    - **Markov chain** – a structure that defines the probability of transitioning from the "current" state to the "next" state.
        - *transition probability* $q(\mathbf{x} \to \mathbf{x}')$ - the probability that the process transitions from state **x** to state **x'**.
        - ***ergodic*** – essentially every state much be reachable from every other and there can be no strictly periodic cycles.
        - *state distribution* $\pi_t(\mathbf{x})$ - the probability of being in state **x** at the *t*-th step of the Markov chain.
    - ***stationary distribution*** – a state distribution such that $\pi_t = \pi_{t+1}$

$$\forall \mathbf{x}' \qquad \pi(\mathbf{x}') = \sum_{\mathbf{x}} \pi(\mathbf{x}) q(\mathbf{x} \to \mathbf{x}')$$

        - This distribution is unique if the chain is *ergodic*.
        - A distribution is stationary if it satisfies the *detailed balance equation:*

$$\forall \mathbf{x}, \mathbf{x}' \qquad \pi(\mathbf{x}) q(\mathbf{x} \to \mathbf{x}') = \pi(\mathbf{x}') q(\mathbf{x}' \to \mathbf{x})$$

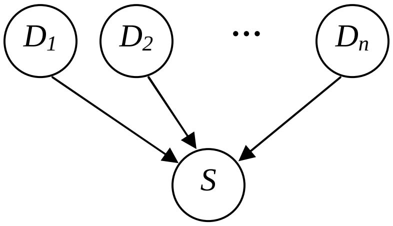
Question 4.12 from Text

IV. Noisy-OR

Suppose there are *n* diseases D_i all of which cause a symptom *S*. In the classical logic world, we might think that if you at least one of the diseases D_i than you would have symptom *S* and you wouldn't have it otherwise. This is modeled by the following logical sentence (a simple OR-gate):

$$S = D_1 \vee D_2 \vee \ldots \vee D_n$$

Of course, we want to incorporate uncertainty into the picture. This is captured by a particular model known as the *Noisy-OR* model. The general graphical structure for this model is simply:



However, this graphical structure does not capture all the intricacies we specified in the logical setting (In fact, the above graphical model is the same for *Noisy-AND* and many other "Noisy" versions of logical gates). The concept of *Noisy-OR* must be captured in the conditional probability table. It must have the following properties:

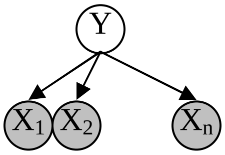
1. We want to model the probabilistic structure of OR such that (roughly) *S=true* if any one of the diseases is present and *S=false* otherwise.
    a. $P(S = true \mid D_1 = D_2 = \ldots = D_n = false) = 0$
    b. $P(S = false \mid D_1 = D_2 = \ldots = D_n = false) = 1$
2. It seems bad form to say there is 0 probability of having a symptom… couldn't there be causes we're not accounting for?
    a. *We assume we've accounted for ALL causes*. Any miscellaneous causes can be captured by an extra **leak node**.
3. Even if a cause (disease) is present, the effect (symptom) might be inhibited. This is the uncertainty we wish to model.
    a. Each cause can be inhibited with probability q_i. Thus,
    $$P(S = false \mid D_1 = D_2 = \ldots = D_n = false, D_i = true) = q_i$$
    b. *We assume each cause is inhibited INDEPENDENTLY*. Thus the probability that we have D_i and D_j but not *S* is given by:
    $$P(S = false \mid D_1 = D_2 = \ldots = D_n = false, D_i = true, D_j = true) = q_i q_j$$
4. Thus, the entire conditional probability table can be fashioned with only *n* parameters $q_1, q_2, \ldots, q_n$ rather than $O(2^n)$.

Note: there is an alternative graphical model that captures these assumptions explicitly through auxiliary variables, but it's not important for our purpose.

V. Alternate Material

Structure of Bayes Nets

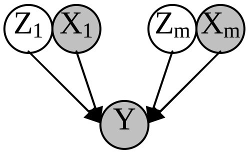
- The structure of a network contains the essential information about the conditional independence of the random variables.
- There are many reoccurring structures that capture common assumptions.
    - Naïve Bayes Model



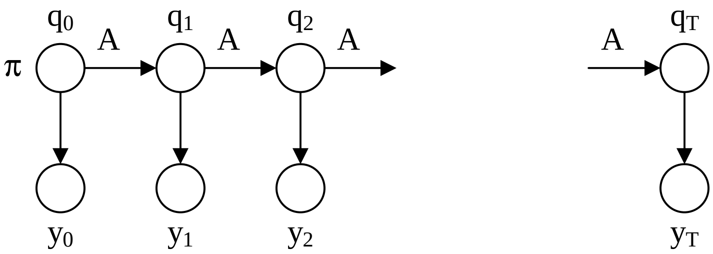
(a) conditionally independent features

    - Noisy Or Model

$$Y = \begin{cases} 1 & \bigvee_{j=1}^{m} \left(X_j \wedge \neg Z_j \right) \\ 0 & otherwise \end{cases}$$



    - Hidden Markov Model



- These models are very important in a branch of AI known as Statistical Machine Learning where we try to learn their parameters from observations of real-world phenomenon we assume follow a given model.
    - Inconsistencies between the exact model are often secondary to the effects captured in the structure of the model.
    - Independence assumptions often don't hold in the real world, but the models still perform well due to the approximate independence exhibited.

Foundations

- ***Conditional Independence*** **–** implies that two variables X,Y are independent given variable Z:

$$P(X,Y \mid Z) = P(X \mid Z)P(Y \mid Z) \qquad P(X \mid Y,Z) = P(X \mid Z)$$

- **Bayes' Rule –** application of product rule that allows diagnostic beliefs to be derived from casual beliefs:

$$P(Y \mid X) = \frac{P(X \mid Y)P(Y)}{P(X)} \qquad P(Y \mid X,e) = \frac{P(X \mid Y,e)P(Y \mid e)}{P(X \mid e)}$$

Chain Rule of Probability Theory – In general,

$$p(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} p(X_i \mid X_1, X_2, \ldots, X_{i-1})$$

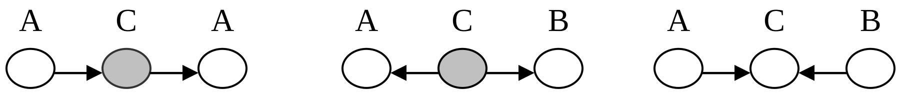
Graphical Model – represents the joint probability distribution over a set of random variables via the independence relationships between those variables, thus concisely encapsulating a family of probability of distributions that respect those independence assumptions.

  - <u>Nodes</u> – correspond in a 1-1 relationship with the variables in the distribution.

- <u>Edges</u> – represent dependence between a pair of random variables. The interpretation of this dependence depends on whether or not the graph is directed.

d-separation – two nodes X and Y in a directed graph are d-separated if every path between X and Y is blocked.

  - A path between X and Y is blocked if it has any of the following 3 cases for any 3 nodes along the path.
    - head-to-tail with intermediary observed: $A \perp\!\!\!\perp B \mid C$
    - tail-to-tail with intermediary observed: $A \perp\!\!\!\perp B \mid C$
    - head to head with neither the intermediary nor any of its descendants observed: $A \perp\!\!\!\perp B \mid \varnothing$



Bayes Ball Algorithm – an algorithm for determining reachability under a particular definition of separation. In particular, it determines if there exists a path from set X_A to set X_B given that the X_C are "specified."

1. Place a ball in all nodes of X_A.
2. For each ball in the graph, explore each direct path the ball could use to move through some neighboring node; this includes return paths where a node serves as both origin and destination. If the path is valid according to the rules of separation, place a ball at the destination.
3. Upon termination, if a ball is in a member of X_B, the set is reachable; return true. Otherwise return false.

Probabilistic Inference – the computation of $P(X_F \mid X_E)$ for a graph $G = (\nu, \varepsilon)$ where $F, E \subseteq \nu$ index sets such that $F \cap E = \varnothing$; disjoint.

- **query nodes**: X_F; we want to obtain the conditional probability of these.
- **evidence nodes**: variables begin conditioned on, X_E
- **remaining nodes**: X_R where $R = \nu \setminus (F \cup E)$. Must be marginalized!
- **marginal** $P(x_F, x_E) = \sum_{x_R} P(x_F, x_E, x_R)$
- **prior** $P(x_E) = \sum_{x_F} P(x_F, x_E)$
- **conditional** $P(x_F \mid x_E) = \dfrac{P(x_F, x_E)}{P(x_E)}$
- Notes:
  - Using the distributive law, factors irrelevant to a summation can be brought outside of it. By associative law, the order of sums can also be swapped.
  - Each summation introduces a new factor that has the marginalized variable removed but incorporates all other variables used in that product.
  - Determining the optimal ordering of sums that minimizes size of intermediate terms is, in general, NP-hard.

- **Conditioning** – the act of basing the probability of the query nodes on specific values of the evidence nodes.
  - **evidence potential** $\delta(x_i, \overline{x}_i)$ - potential that is 1 if $x_i = \overline{x}_i$; 0 otherwise: Kronecker delta function.
  - evidence potentials transform evaluations into sums:

$$g(\overline{x}_i) = \sum_{x_i} g(x_i)\,\delta(x_i, \overline{x}_i)$$

- Continuous Random Variables:
  - **discretization** – dividing variable's possible values into intervals.
  - **parameterization** – describing the variable's distribution by a finite set of parameters.
  - **hybrid BN** – a BN containing both discrete and continuous variables.
  - conditional distributions for continuous variables:
    - discrete parents' values are enumerated.
    - continuous parents' must be summarized in a distribution, for instance, the linear Gaussian distribution where mean varies linearly with parents' value and std dev is fixed: $\mu = ax + b$.
    - linear Gaussian has joint distribution is multivariate Gaussian over all variables. These are combined with discrete variables in conditional Gaussians.
  - conditional distributions for discrete variables with continuous parents.

Approximate Inference in Bayesian Networks

- Monte Carlo algorithms – algorithms that approximate a desired quantity through random sampling.
- Direct Sampling
- Rejection Sampling
- Likelihood Weighting
  -

15: Probabilistic Reasoning Over Time

Modeling Uncertainty over Time

- Setting
  - X_t - a set of unobserved state variables at time *t*.
  - E_t - a set of observable evidence variables for time *t*.
  - *a:b* – denotes an interval from *a* to *b*.
- **Stationary Process** – process of change that is governed by laws that do not change over time.
- **Markov Assumption** – current state depends only on a *finite* history of previous states. Processes satisfying this assumption are *Markov Processes (Chains)*.
  - **transition model** – law describing how state changes over time.

    $$P\left(X_t \mid X_{0:t-1}\right) = P\left(X_t \mid X_\alpha\right) \text{ where } \alpha \subseteq \{1 \ldots t-1\}$$
  - **first-order Markov Process** – current state is solely dependent on the previous state
    - transition model: $P\left(X_t \mid X_{t-1}\right)$
- We assume the evidence variables at time *t* depend only on the current state.
  - **sensor model** – law describing how the evidence depends on the state.

    $$P\left(E_t \mid X_{0:t}, E_{0:t-1}\right) = P\left(E_t \mid X_t\right)$$
- prior probability for the initial state: $P\left(X_0\right)$
- complete joint

  $$P\left(X_{0:T}, E_{1:T}\right) = P\left(X_0\right) \prod_{t=1}^{T} P\left(X_t \mid X_{t-1}\right) P\left(E_t \mid X_t\right)$$
- Ways to deal with inaccurate Markov modeling:
  1. Increase the order of the Markov process
  2. Increase the set of state variables

Filter (monitoring) – the task of computing the *belief state* – the posterior distribution of the current state given all evidence; $P\left(X_T \mid e_{1:T}\right)$.

- Recursive estimation – forward chaining.

  $$P\left(X_t \mid e_{1:t}\right) \propto P\left(e_t \mid X_t\right) \sum_{X_{t-1}} P\left(X_t \mid X_{t-1}\right) \underbrace{P\left(X_{t-1} \mid e_{1:t-1}\right)}_{\text{recursive estimate}}$$

  $$f_{1:t} \propto FORWARD\left(f_{1:t-1}, e_t\right)$$
- When the state variables are discrete, this update is constant in space and time.
- *Likelihood* $P\left(e_{1:T}\right)$ can be calculated by a likelihood message: $l_{1:t} = P\left(X_t, e_{1:t}\right)$:

  $$L_{1:T} = \sum_{X_T} l_{1:T}\left(X_T, e_{1:T}\right)$$

Prediction – task of computing the posterior distribution over a *future* state, given all evidence; $P\left(X_{T+k} \mid e_{1:T}\right)$ where *k > 0*.

- This is equivalent to filtering without new evidence. Hence, we can easily derive the following update:

$$P\left(X_{T+k} \mid e_{1:T}\right) = \sum_{X_{t+k}} P\left(X_{T+k} \mid X_{T+k-1}\right) \underbrace{P\left(X_{T+k-1} \mid e_{1:T}\right)}_{\text{recursive estimate}}$$

- **stationary distribution –** The fixed point of the Markov process that is approached upon successive applications of the transition model.
  - **mixing time –** the amount of time required to reach stationarity.
  - Prediction is doomed to failure for future times more than a small fraction of the mixing time.

Smoothing (hindsight) – task of computing posterior distribution for a *past* state, given all evidence; $P\left(X_k \mid e_{1:T}\right)$ where $0 \le k < T$.

- Accounting for hindsight is done with an additional backwards message:

$$P\left(X_k \mid e_{1:T}\right) \propto \underbrace{P\left(X_k \mid e_{1:k}\right)}_{f_{1:k}} \underbrace{P\left(e_{k+1:T} \mid X_k\right)}_{b_{k+1:T}}$$

$$b_{k+1:T} = \sum_{X_{k+1}} P\left(e_{k+1} \mid X_{k+1}\right) P\left(X_{k+1} \mid X_k\right) b_{k+2:T}$$

- The time and space needed for each backward message are constant.
- Thus, the process of smoothing with respect to $e_{1:T}$ is *O(t)*.
- Thus, to smooth the whole sequence naively, requires $O(t^2)$.
- using dynamic programming the cost is only *O(t)* by recording results of forward filtering over the entire sequence while running the backward algorithm from *T* to 1 and use the smoothed message at each time step ➔ **forward-backward algo**.
  - space is now $O\left(|f|t\right)$
- In on-line setting, smoothed estimates must be computed for earlier time slices as new observations are added:
  - **fixed-lag smoothing –** smoothing is done for the time slice *d* steps behind the current time *T*.

Most Likely Explanation – task of finding the sequence of states most likely to have generated a sequence of observations; $\arg\max_{x_{1:t}} P(x_{1:t} \mid e_{1:t})$.

- most likely sequence must consider joint probabilities over all time steps.
- *there is a recursive relationship between most likely paths to each state X_{t+1} and the most likely paths to each state X_t.*
- Recursive formulation:

$$\max_{X_{1:t-1}} P(X_{1:t} \mid e_{1:t}) \propto \underbrace{P(e_t \mid X_t)}_{\textit{observation}} \max_{X_{t-1}} \left[\underbrace{P(X_t \mid X_{t-1})}_{\textit{transition}} \underbrace{\max_{X_{1:t-2}} P(X_{1:t-1} \mid e_{1:t-1})}_{\text{previous message}} \right]$$

    - messages: $m_{1:t} = \max_{X_{1:t-1}} P(X_{1:t} \mid e_{1:t})$
    - summation over X_t replaced by a maximization.
- Pointers are used to retrieve the most-likely explanation
- Viterbi algorithm has a space and time requirement of *O(t)*.

Learning – task of learning the transition and sensor models from observed data. This process leverages inference through EM.

Hidden Markov Models (HMM) – a temporal probabilistic model in which the state of the process is described by a *single discrete* random variable and transitions obey the Markov assumption.

- transition model: $T_{ij} = P(X_t = j \mid X_{t-1} = i)$
- observation model: $(\mathbf{O_t})_{i,i} = P(e_t \mid X_t = i)$
    - *forward* message - $\mathbf{f}_{1:t+1} \propto \mathbf{O}_{t+1}\mathbf{T}^T\mathbf{f}_{1:t}$
    - *backward* message - $\mathbf{b}_{k+1:t} \propto \mathbf{T}\mathbf{O}_{k+1}\mathbf{b}_{k+2:t}$
    - time complexity of forward-backward becomes $O(S^2 t)$ where S is the number of hidden states and space complexity is $O(St)$.

Kalman Filters – a temporal probabilistic model for continuous state spaces under the Markov assumption and using linear Gaussian distributions to model the states. A Kalman filter can model any system of continuous state variables with noisy measurements.

- a *multivariate Gaussian* distribution can be specified completely by its mean $\boldsymbol{\mu}$ and its covariance matrix $\boldsymbol{\Sigma}$.
- In general, filtering with continuous or hybrid spaces generate state distributions whose representations grow without bound, but the Gaussian distribution is "well-behaved" since it has the following properties:
  1. If the current distribution $P\left(\mathbf{X}_t \mid \mathbf{e}_{1:t}\right)$ is Gaussian and the transition model $P\left(\mathbf{X}_{t+1} \mid \mathbf{x}_t\right)$ is linear Gaussian, then the predicted distribution of the next step is:
  $$P\left(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t}\right) = \int_{\mathbf{x}_t} P\left(\mathbf{X}_{t+1} \mid \mathbf{x}_t\right) P\left(\mathbf{x}_t \mid \mathbf{e}_{1:t}\right) d\mathbf{x}_t$$
  2. If the predicted distribution is Gaussian and the observation (sensor) model is linear Gaussian, then conditioning on new evidence yields the updated distribution:
  $$P\left(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t+1}\right) \propto P\left(\mathbf{e}_{1:t+1} \mid \mathbf{X}_{t+1}\right) P\left(\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t}\right)$$
- General formulation:
  $$P\left(\mathbf{x}_{t+1} \mid \mathbf{x}_t\right) = N\left(\mathbf{F}\mathbf{x}_t, \boldsymbol{\Sigma}_x\right)\left(\mathbf{x}_{t+1}\right)$$
  - $\mathbf{F}$ and $\boldsymbol{\Sigma}_x$ describe the linear transition model & noise.
  $$P\left(\mathbf{z}_t \mid \mathbf{x}_t\right) = N\left(\mathbf{H}\mathbf{x}_t, \boldsymbol{\Sigma}_z\right)\left(\mathbf{z}_t\right)$$
  - $\mathbf{H}$ and $\boldsymbol{\Sigma}_z$ describe the linear sensor model & noise.
- Updates:
  $$\boldsymbol{\mu}_{t+1} = \mathbf{F}\boldsymbol{\mu}_t + \mathbf{K}_{t+1}\left(\mathbf{z}_{t+1} - \mathbf{H}\mathbf{F}\boldsymbol{\mu}_t\right)$$
  $$\boldsymbol{\Sigma}_{t+1} = \left(\mathbf{I} - \mathbf{K}_{t+1}\right)\left(\mathbf{F}\boldsymbol{\Sigma}_t\mathbf{F}^T + \boldsymbol{\Sigma}_x\right)$$
  - Kalman gain $K_{t+1} = \left(\mathbf{F}\boldsymbol{\Sigma}_t\mathbf{F}^T + \boldsymbol{\Sigma}_x\right)\mathbf{H}^T\left(\mathbf{H}\left(\mathbf{F}\boldsymbol{\Sigma}_t\mathbf{F}^T + \boldsymbol{\Sigma}_x\right)\mathbf{H}^T + \boldsymbol{\Sigma}_z\right)^{-1}$
    - A measure of "how seriously to take the new observation" relative to the prediction.
  - predicted state at t+1 is $\mathbf{F}\boldsymbol{\mu}_t$, predicted observation is $\mathbf{H}\mathbf{F}\boldsymbol{\mu}_t$, and error of predicted observation is $\left(\mathbf{z}_{t+1} - \mathbf{H}\mathbf{F}\boldsymbol{\mu}_t\right)$.
- Extended Kalman Filter (EKF) – allows for limited nonlinearity in the model by modeling the system *locally* as linear in $\mathbf{x}_t$ in the region of $\mathbf{x}_t = \boldsymbol{\mu}_t$.
- Switching Kalman Filter –