6.4110 PGM PSS

Recitation: HMMs, Kalman Filters, and Inference

Feb 20th

Overview

Definitions

Distribution vs. Random Variable

- **High-level:** A distribution is a rule; a random variable (RV) is a value drawn from it.
- **Notation:** $A \sim \mathcal{N}(\mu, \sigma^2)$
- Note: $\sim$ is **not** an equals sign.

A random variable is not the same object as its distribution.

Summing Random Variables

If $A \sim \mathcal{N}(\mu_1, {\sigma_1}^2)$ and $C \sim \mathcal{N}(\mu_2, {\sigma_2}^2)$ are independent:

Convolution (Sum of Values)

$$A + C \sim \mathcal{N}(\mu_1 + \mu_2, {\sigma_1}^2 + {\sigma_2}^2)$$

Adding random variables **increases** uncertainty (variance adds).

Warning: Summing Distributions

$$\mathcal{N}(\mu_1, {\sigma_1}^2) + \mathcal{N}(\mu_2, {\sigma_2}^2)$$

This is **NOT** a single Gaussian. It is a *mixture model* (bimodal) and needs normalization.

Scaling and Multiplying

Scaling

For a constant c:

$$cA \sim \mathcal{N}\left(c\mu_1, (c)^2{\sigma_1}^2\right)$$

Product of PDFs (Inference)

Multiplying two Gaussian **PDFs** (information fusion) yields another Gaussian (scaled):

$$\mathcal{N}(\mu_1, {\sigma_1}^2) \times \mathcal{N}(\mu_2, {\sigma_2}^2) \propto \mathcal{N}(\mu^*, (\sigma^*)^2)$$

Warning: Product of Random Variables

Even though the product of two Gaussian *distributions* is Gaussian, the product of two Gaussian *random variables* $A \times C$ is **NOT** Gaussian!

Product of Gaussians: The Math

The resulting variance shrinks (precision adds):

$$\frac{1}{(\sigma^*)^2} = \frac{1}{{\sigma_1}^2} + \frac{1}{{\sigma_2}^2}$$

The resulting mean is a precision-weighted average:

$$\mu^* = (\sigma^*)^2 \left(\frac{\mu_1}{{\sigma_1}^2} + \frac{\mu_2}{{\sigma_2}^2} \right)$$

Implication: This property enables efficient exact inference in Factor Graphs and Kalman Filters.

Multivariate Definition

Vector $\vec{x} = [x_1, \ldots, x_n]^T$.

$$\vec{x} \sim \mathcal{N}(\vec{\mu}, \mathbf{\Sigma})$$

Partition variables into sets A and B:

$$\vec{x} = \begin{bmatrix} \vec{x}_A \\ \vec{x}_B \end{bmatrix}, \quad \vec{\mu} = \begin{bmatrix} \vec{\mu}_A \\ \vec{\mu}_B \end{bmatrix}, \quad \mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma}_{AA} & \mathbf{\Sigma}_{AB} \\ \mathbf{\Sigma}_{BA} & \mathbf{\Sigma}_{BB} \end{bmatrix}$$

Marginalization vs. Conditioning

Marginalization (Dropping dims)

To find $P(\vec{x}_A)$, ignore B:

$$\vec{x}_A \sim \mathcal{N}(\vec{\mu}_A, \mathbf{\Sigma}_{AA})$$

Conditioning (Slicing)

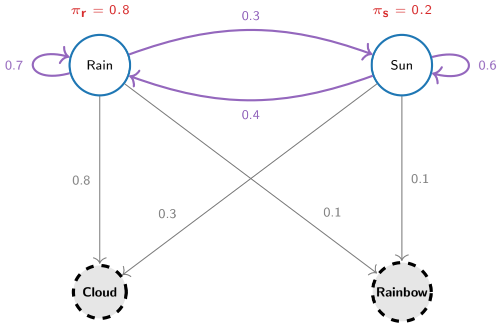
Use observation $\vec{x}_B = \vec{b}$:

$$\vec{\mu}_{A|B} = \vec{\mu}_A + \mathbf{\Sigma}_{AB}\mathbf{\Sigma}_{BB}^{-1}(\vec{b} - \vec{\mu}_B)$$

$$\mathbf{\Sigma}_{A|B} = \mathbf{\Sigma}_{AA} - \mathbf{\Sigma}_{AB}\mathbf{\Sigma}_{BB}^{-1}\mathbf{\Sigma}_{BA}$$

Note: Conditioning *reduces* uncertainty (covariance subtracts positive definite term).

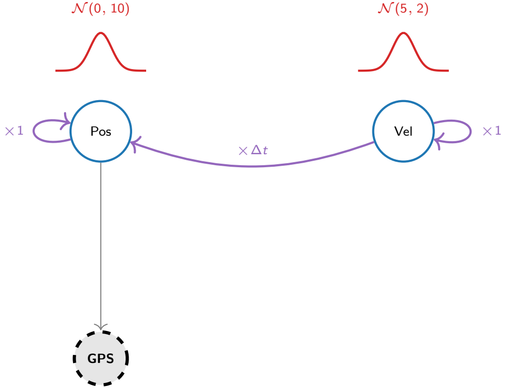
Discrete HMM ($N = 2$ States)



- **States:** Finite set (Rain, Sun).
- **Transitions:** Probabilities $P(j|i)$.
- **Emissions:** Probabilities $P(y|i)$.

2. Continuous Dynamics (LG-HMM)

Linear-Gaussian HMM ($N = 2$ Dims)



- **States:** Continuous vector.
- **Transitions:** Linear weights.
- **Emissions:** Linear + Noise.
- $\vec{x}_t = \mathbf{A}\vec{x}_{t-1} + \text{noise}$

Overview: Forward-Backward

Goal: Compute $P(\vec{z}_t | \vec{O}_{0:T})$ (marginal probability of state at time t given ALL observations).

1. **Filtering (Forward):** Where am I now, given past history?
2. **Smoothing (Backward):** Given that I ended up *there*, where was I likely to be before?

This is **Message Passing** on a chain.

The Algorithms

1. Forward (Filtering) α

$P(\vec{z}_t | \vec{O}_{1:t})$

$$\vec{\alpha}_t \propto \underbrace{\mathbf{B}_{\vec{O}_t}}_{\text{Local Evidence}} \cdot (\underbrace{\mathbf{A}^T}_{\text{Transition}} \underbrace{\vec{\alpha}_{t-1}}_{\text{Msg from Past}})$$

Base case: $\vec{\alpha}_0 = \vec{\pi}$

2. Backward (Smoothing) β

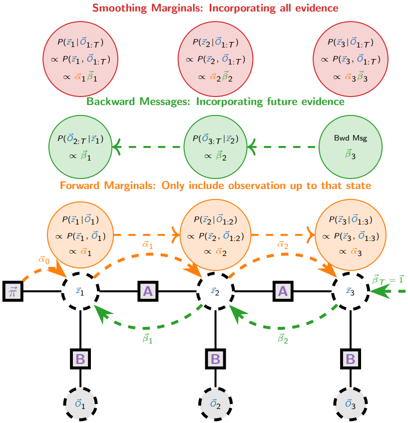
Messages from the future needed for $P(\vec{z}_t | \vec{O}_{1:T})$.

$$\vec{\beta}_t = \underbrace{\mathbf{A}}_{\text{Transition}} (\underbrace{\vec{\beta}_{t+1}}_{\text{Msg from Future}} \cdot \underbrace{\mathbf{B}_{\vec{O}_{t+1}}}_{\text{Future Evidence}})$$

Base case: $\vec{\beta}_{T+1} = \mathbf{1}$

Result: Marginal $\propto \vec{\alpha}_t \cdot \vec{\beta}_t$.

Visualizing Inference (Factor Graph)



Viterbi Algorithm

Problem: Finding the most likely state at each step independently can lead to invalid paths.
Goal: Find the single global sequence $\vec{z}^{*}_{1:T}$ that maximizes $P(\vec{z}_{1:T}|\vec{O}_{1:T})$.

Max-Product Recursion

Replace summation with maximization.

$$\delta_t(j) = \underbrace{\mathbf{B}_{\vec{O}_t}[j]}_{\text{Evidence}} \cdot \max_i \left(\underbrace{\mathbf{A}_{ij}}_{\text{Transition}} \cdot \delta_{t-1}(i) \right)$$

Backtracking: Store pointers $\psi_t(j)$ to the best previous state i. Trace back from the end.

Comparison

Feature	Discrete HMM	Linear-Gaussian HMM
State Space	N discrete categories	Continuous vector $\mathbb{R}^n$
Belief Form	Categorical Vector (sums to 1)	Gaussian $\mathcal{N}(\vec{\mu}, \mathbf{\Sigma})$
Dynamics	Matrix Multiplication	Linear Map $\mathbf{A}\vec{x} +$ noise
Inference	Grid / Matrix Ops	Kalman Filter (Closed form)

The graphical model structure is identical; only the probability tables change to density functions.

Bayes' Rule: Univariate Case

Prior: $x \sim \mathcal{N}(\mu_0, \sigma_0^2)$ **Likelihood:** $y \mid x \sim \mathcal{N}(x, \sigma_y^2)$
The posterior is a weighted average based on **precision** $\lambda = 1/\sigma^2$.

Posterior Mean

$$\mu_{post} = \frac{\lambda_0 \mu_0 + \lambda_y y}{\lambda_0 + \lambda_y}$$

We trust the source with higher precision (lower variance) more.

Posterior Precision

$$\lambda_{post} = \lambda_0 + \lambda_y$$

Information always adds up; uncertainty always decreases.

Bayes' Rule: Multivariate Case

Prior: $\vec{x} \sim \mathcal{N}(\vec{\mu}_0, \mathbf{\Sigma}_0)$ **Obs:** $\vec{y} = \mathbf{A}\vec{x} + \mathbf{b} + \eta$

Update Equations

$$\mathbf{\Sigma}_{post}^{-1} = \mathbf{\Sigma}_0^{-1} + \mathbf{A}^T \mathbf{\Sigma}_y^{-1} \mathbf{A}$$

$$\mathbf{\Sigma}_{post}^{-1} \vec{\mu}_{post} = \mathbf{\Sigma}_0^{-1} \vec{\mu}_0 + \mathbf{A}^T \mathbf{\Sigma}_y^{-1} (\vec{y} - \vec{b})$$

This is the rigorous math behind the **Kalman Filter update step**.

Framework Summary

Framework	Vars	Belief
CSP	Disc	Boolean (Valid/Invalid)
Discrete BN	Disc	Probability Dist. across domain for each variable.
Gaussian BN	Cont	Gaussian distribution per variable.
Discrete HMM	Disc	Probability Dist. across domain for single variable over time.
Linear Gaussian HMM	Cont	Gaussian per state dimension (μ, Σ)

Factor Graphs

A **unifying representation** for all the above. They graph the variables and the local functions (factors) that connect them.

The Process

Concept: An online recursive cycle:

1. **Predict:** Using physics/dynamics.
2. **Measure:** Get noisy sensor data.
3. **Update:** Fuse them based on trust (Covariance).

Kalman Gain (K): The "slider" between current belief and new data.

- High Sensor Noise $\rightarrow$ Low **K** (Ignore data).
- Low Sensor Noise $\rightarrow$ High **K** (Jump to data).

The Equations

1. Predict (Time Update)

$$\vec{\mu}_{t|t-1} = \mathbf{A}\vec{\mu}_{t-1}$$
$$\boldsymbol{\Sigma}_{t|t-1} = \mathbf{A}\boldsymbol{\Sigma}_{t-1}\mathbf{A}^T + \mathbf{Q}$$

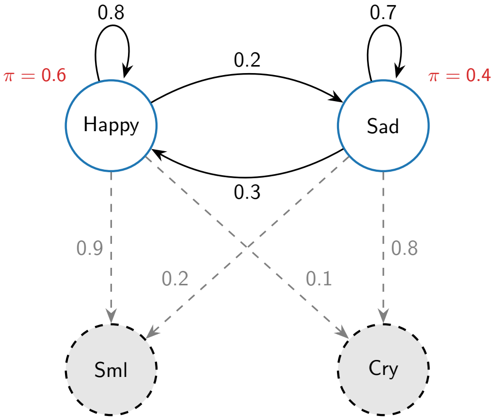
2. Update (Measurement Update)

$$\mathbf{K}_t = \boldsymbol{\Sigma}_{t|t-1}\mathbf{H}^T(\mathbf{H}\boldsymbol{\Sigma}_{t|t-1}\mathbf{H}^T + \mathbf{R})^{-1}$$
$$\vec{\mu}_t = \vec{\mu}_{t|t-1} + \mathbf{K}_t(\vec{y}_t - \mathbf{H}\vec{\mu}_{t|t-1})$$
$$\boldsymbol{\Sigma}_t = (\mathbf{I} - \mathbf{K}_t\mathbf{H})\boldsymbol{\Sigma}_{t|t-1}$$

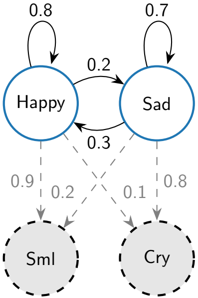
Example Setup

Problem Setup:

- States: Happy (H), Sad (S).
- Observations: Smile (Sml), Cry (C).



Example Q1: Define Matrices



Model Matrices

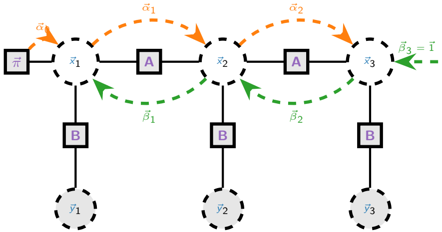
Transition A (H, S): $\begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \end{bmatrix}$

Emission B (Sml, Cry): $\begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix}$

Q1: Get Transition & Emission Matrices

1. **Prior:** $\vec{\pi} = [0.6, 0.4]$
2. **Transition A:**
   $A_{ij} = P(\vec{x}_{t+1} = j | \vec{x}_t = i)$
3. **Emission B:**
   $B_{ik} = P(\vec{y}_t = k | \vec{x}_t = i)$

Q2: Factor Graph



Model Matrices

A: $\begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \end{bmatrix}$ **B**: $\begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix}$

Evidence

Sequence: $\vec{y}_1 = \text{Sml}$, $\vec{y}_2 = \text{Cry}$, $\vec{y}_3 = \text{Sml}$

Tip: Unroll the HMM over time.

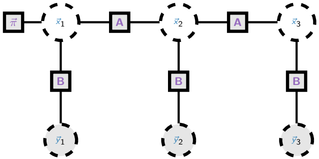
Q3: Forward Algorithm

Model Matrices

A: $\begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \end{bmatrix}$ **B**: $\begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix}$

Question

Compute the forward messages $\vec{\alpha}_1$, $\vec{\alpha}_2$, and $\vec{\alpha}_3$ given the sequence $\vec{y}_{1:3} = (\text{Sml}, \text{Cry}, \text{Sml})$.
What is the most likely state at $t = 3$ given all observations up to that point?



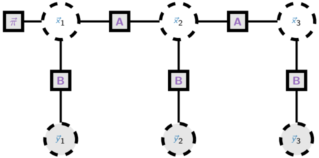
Q4: Backward Algorithm

Model Matrices

A: $\begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \end{bmatrix}$ **B**: $\begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix}$

Question

Compute the backward messages $\vec{\beta}_3$, $\vec{\beta}_2$, and $\vec{\beta}_1$ given the sequence $\vec{y}_{1:3} = (\text{Sml}, \text{Cry}, \text{Sml})$.
What is the probability of the state at $t = 1$ given all observations $\vec{y}_{1:3}$?



Q5: Predict Next Observation

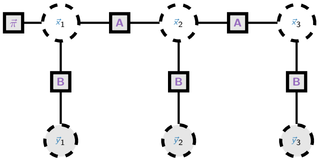
Model Matrices

A: $\begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \end{bmatrix}$ **B**: $\begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix}$

Question

Using the results from Forward Algorithm at $t = 3$ ($P(\vec{x}_3|\vec{y}_{1:3})$), predict the probability of the next observation:

What is the $P(\vec{y}_4 = \text{Cry}|\vec{y}_{1:3})$?



Q6: Max Path via Enumeration

Model Matrices

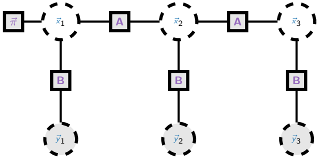
A: $H, S \to H, S = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \end{bmatrix}$

B: $Sml, Cry = \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix}$

$\vec{\pi}$: $H, S = [0.6, 0.4]$

Question

What is the most likely path $\vec{x}_{1:3}$ manually evaluated by listing out and computing all $2^3 = 8$ paths?



Q7: Viterbi Path

Model Matrices

A: $\begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \end{bmatrix}$ **B**: $\begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix}$

Question

What is the most likely path $\vec{x}_{1:3}$ using the Viterbi (**Max-Sum**) Algorithm?