DS-GA 3001.009
Modeling Time Series Data
Lab 4

Artie Shen | Center for Data Science

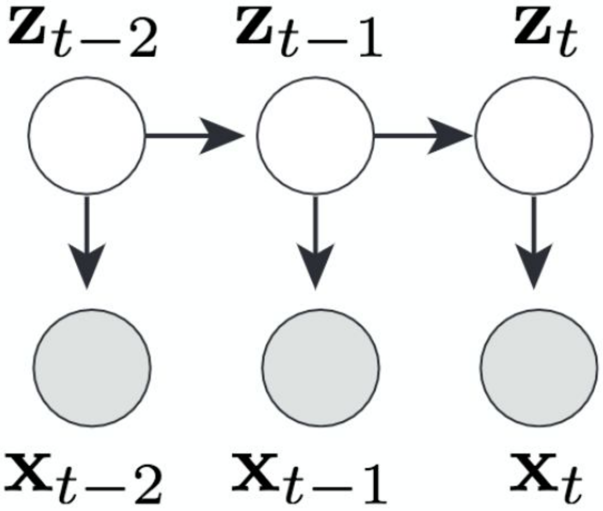
NYU

Agenda

- Recap
  - Kalman Filter
  - General EM

- Programming
  - Sampling
  - Filtering
  - Smoothing
  - Learning

Formulation



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:

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

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

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

Filtering

Goal $P(z_n|x_1, ..., x_n) \sim N(\mu_n, V_n)$

Prediction Step $\mu_n^{pred} = A\mu_{n-1},\ P_{n-1} = AV_{n-1}A^T + Q$

Observation Step

- Innovation $\widetilde{x} = x_n - C\mu_n^{pred}$
- Innovation Co-variance $\widetilde{R} = CP_{n-1}C^T + R$
- Kalman Gain $K = P_{n-1}C^T\widetilde{R}^{-1}$

Update Step $\mu_n = \mu_n^{pred} + K\widetilde{x}, V_n = (I - KC)V_n^{pred}$

Smoothing

Goal $P(z_n|x_1, ..., x_n, x_{n+1}, ..., x_N) \sim N(\widehat{\mu}_n, \widehat{V}_n)$

Forward Path Run the filtering from $i = 0$ to N to obtain $(\mu_i, V_i), \forall i \in [1, N]$.

Backward Path

- iterate backward from N to 1
- initialization, $\widehat{\mu}_N = \mu_N, \widehat{V}_N = V_N$
- $J_n = V_n A^T P_n^{-1}$
- $\widehat{\mu}_n = \mu_n + J_n(\widehat{\mu}_{n+1} - A\mu_n)$
- $\widehat{V}_n = V_n + J_n(\widehat{V}_{n+1} - P_n)J_n^T$

NYU

Generalized EM

Let $q(z)$ be some distribution over z, then $ln[p(x|\theta)] = \sum_z q(z)ln[p(x|\theta)]$.

$$
\begin{aligned}
ln[p(x|\theta)] &= \sum_z q(z)ln[p(x|\theta)] \\
&= \sum_z q(z)ln[\frac{p(x,z|\theta)}{p(z|x,\theta)}] \\
&= \sum_z q(z)ln[\frac{p(x,z|\theta)/q(z)}{p(z|x,\theta)/q(z)}] \\
&= \sum_z q(z)ln[\frac{p(x,z|\theta)}{q(z)}] - \sum_z q(z)ln[\frac{p(z|x,\theta)}{q(z)}] \\
&= L(q,\theta) + KL(q||p)
\end{aligned}
$$

Generalized EM

We have following observations:

- $L(q, \theta) = \sum_z q(z) ln[\frac{p(x,z|\theta)}{q(z)}]$ only depends on θ because we marginalize over z.

- $KL(q||p) = -\sum_z q(z) ln[\frac{p(z|x,\theta)}{q(z)}]$ is the Kullback-Leibler divergence. $KL(q||p) \geq 0$. $KL(q||p) = 0$ only when $q(z)$ matches the posterior distribution $p(z|x, \theta)$.

- $L(q, \theta) \leq ln[p(x|\theta)]$, the equal sign holds when $KL(q||p) = 0$.

E step

- To make sure our lower bound $L(q, \theta)$ reflects the true log likelihood $ln[p(x|\theta)]$, we need to make $KL(q||p) = 0$.
- This implies that we need to update z so that $q(z) = p(z|x, \theta)$ given a fixed set of θ.

M step

- Since $q(z)$ is updated, we keep it fixed.We need to find new θ so that it maximizes the log likelihood: $\theta^* = argmax_\theta L(q, \theta)$
- In the E step, we have $q(z) = p(z|x, \theta^{old})$. If we substitute $q(z)$ in $L(q, \theta)$, we can further reduce:
  $L(q, \theta) = \sum_z p(z|x, \theta^{old}) ln[p(x, z|\theta)] - p(z|x, \theta^{old}) ln[p(z|x, \theta^{old})]$.
- Note that the latter term $p(z|x, \theta^{old}) ln[p(z|x, \theta^{old})]$ denotes the entropy of the $p(z|x, \theta^{old})$ and is independent of θ which we need to update.
- Therefore, we only need to focus on $Q(\theta, \theta^{old}) = \sum_z p(z|x, \theta^{old}) ln[p(x, z|\theta)]$.

Learning Kalman Filter

- Complete-data log likelihood:
$ln[p(X, Z|\theta)] = ln[p(z_1|\mu_0, V_0)] + \sum_{n=2}^{N} ln[p(z_n|z_{n-1}, A, Q)] + \sum_{n=1}^{N} ln[p(x_n|z_n, C, R)]$

- Run Kalman Smoothing (E step) and pre-compute the "ingredients":
$E[z_n] = \widehat{\mu}_n,\ E[z_n z_{n-1}^T] = J_{n-1}\widehat{V}_n + \widehat{\mu}_n \widehat{\mu}_{n-1}^T,\ E[z_n z_n^T] = \widehat{V}_n + \widehat{\mu}_n \widehat{\mu}_n^T$

- Update initial states:
$\mu_0 = E[z_1], V_0 = E[z_1 z_1^T] - E[z_1]E[z_1]^T$

- Update Transition matrix / covariance:
$A = (\sum_{n=2}^{N} E[z_n z_{n-1}^T])(\sum_{n=2}^{N} E[z_{n-1} z_{n-1}^T])^{-1}$
$Q = \frac{1}{N-1} \sum_{n=2}^{N} E[z_n z_n^T] - AE[z_{n-1} z_n^T] - E[z_n z_{n-1}^T]A + AE[z_{n-1} z_{n-1}^T]A^T$

- Update Emission matrix / covariance:
$C = (\sum_{n=1}^{N} x_n E[z_n^T])(E[z_n z_n^T])^{-1}$
$R = \frac{1}{N} \sum_{n=1}^{N} x_n x_n^T - CE[z_n]x_n^T - x_n E[z_n^T]C + CE[z_n z_n^T]C^T$

Programming

- Github:
  - https://github.com/charlieblue17/timeseries2018
- pip install pykalman
- Due Date 03/01/2018 06:45 pm on NYU Classes
- Please rename your submission to **net_id.ipynb**