Homework 4
M1522.001300 Probabilistic Graphical Models (2016 Fall)

2015-21259 Hyunwoo Lee

Date: November 8 Tuesday

1 Kalman Filtering

1. What is the purpose of Kalman Filtering? Write down the quantity that Kalman Filtering estimates.

   An the first time, the purpose of Kalman Filtering is to remove the noise from the information or to extract the required information.

   There are two kinds of estimates that is infered from this graphical model. One is the filtering and another is the smoothing.

   The filtering is to estimate the current hidden state based on the sequence of observations, i.e. $P(x_t|y_0, \cdots, y_t)$.

   The smoothing is to estimate the hidden state at the specific past t based on the sequence of observations up to $T\,(t < T)$, i.e. $P(x_t|y_0, \cdots, y_T)$.

2. Write down the equations for Predict Step and Update Step for Kalman Filtering

   1) Predict step

   $$\begin{aligned} P(x_t|y_{0|t}) &\rightarrow P(x_{t+1}|y_{0|t}) \\ P(x_{t+1}|y_{0|t}) &\rightarrow P(x_{t+1}|y_{0|t+1}) \end{aligned}$$

   Because the Kalman Filtering is over multivariate Gaussian,
   we just need to know the mean and the covariance.

   From the dynamic model, $x_{t+1} = Ax_t + Gw_t$

   $\widehat{x}_{t+1|t} = E(x_{t+1}|y_1, \cdots, y_t) = A\widehat{x}_{t|t}$
   $P_{t+1|t} = E((x_{t+1} - \widehat{x}_{t+1|t})(x_{t+1} - \widehat{x}_{t+1|t})^T|y_1, \cdots, y_t)$
   $\quad = AP_{t|t}A^T + GQG^T$

   2) Update step

   Because $P(x_{t+1}, y_{t+1}) \sim N(m_{t+1}, V_{t+1})$,

   $$m_{t+1} = \begin{pmatrix} \widehat{x}_{t+1|t} \\ C\widehat{x}_{t+1|t} \end{pmatrix}, \quad V_{t+1} = \begin{pmatrix} P_{t+1|t} & P_{t+1|t}C^T \\ CP_{t+1|t} & CP_{t+1|t}C^T + R \end{pmatrix}$$

   This is because, for $V_{t+1|t}$,

   $Cov(X, X) = P_{t+1|t}$

   $Cov(Y, Y) = E((y_{t+1} - \widehat{y}_{t+1|t})(y_{t+1} - \widehat{y}_{t+1|t})^T|y_1, \cdots, y_t)$
   $\quad = E((Cx_{t+1} + v_{t+1} - C\widehat{x}_{t+1|t})(Cx_{t+1} + v_{t+1} - C\widehat{x}_{t+1|t})^T|y_1, \cdots, y_t)$
   $\quad = CP_{t+1|t}C^T + R$

$$Cov(X,Y) = E((x_{t+1} - \widehat{x}_{t+1|t})(y_{t+1} - \widehat{y}_{t+1|t})^T | y_1, \cdots, y_t) = CP_{t+1|t}$$

$$Cov(Y,X) = E((y_{t+1} - \widehat{y}_{t+1|t})(x_{t+1} - \widehat{x}_{t+1|t})^T | y_1, \cdots, y_t) = P_{t+1|t}C^T$$

Now, we get

$$\begin{aligned}
\widehat{x}_{t+1|t+1} &= E(x_{t+1}|y_1, \cdots, y_t) \\
&= \widehat{x}_{t+1|t} + P_{t+1|t}C^T(CP_{t+1|t}C^T + R)^{-1}(y_{t+1} - C\widehat{x}_{t+1|t}) \\
&= \widehat{x}_{t+1|t} + K_{t+1}(y_{t+1} - C\widehat{x}_{t+1|t})
\end{aligned}$$

$$\begin{aligned}
P_{t+1|t+1} &= P_{t+1|t} - P_{t+1|t}C^T(CP_{t+1|t}C^T + R)^{-1}CP_{t+1|t} \\
&= P_{t+1|t} - K_{t+1}CP_{t+1|t}
\end{aligned}$$

$$(\, K_{t+1} \triangleq P_{t+1|t}C^T(CP_{t+1|t}C^T + R)^{-1}\,)$$

3. Analyze the time complexity for Kalman Filtering algorithm, and answer in big-O notations. In this question, assume that $x_t \in \mathbb{R}^n$ and $y_t \in \mathbb{R}^m$.

   Kalman Filter algorithm can be written as:

$$\begin{aligned}
\widehat{x}_{t+1|t} &= A\widehat{x}_{t|t} \\
P_{t+1|t} &= AP_{t|t}A^T + GQG^T \\
\widehat{x}_{t+1|t+1} &= \widehat{x}_{t+1|t} + K_{t+1}(y_{t+1} - C\widehat{x}_{t+1|t}) \\
P_{t+1|t+1} &= P_{t+1|t} - K_{t+1}CP_{t+1|t} \\
(\, K_{t+1} &\triangleq P_{t+1|t}C^T(CP_{t+1|t}C^T + R)^{-1}\,)
\end{aligned}$$

   Now let see the time complexity for each.

   1) $\widehat{x}_{t+1|t} = A\widehat{x}_{t|t}$

      Because $\widehat{x}_{t|t}, \widehat{x}_{t+1|t} \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n\times n}$,
      to calculate $A\widehat{x}_{t|t}$, there needs n multiplications with $n-1$ additions for each element in $\widehat{x}$, e.g. $\widehat{x}_{t+1,1} = \widehat{x}_{t,1} \cdot a_{1,1} + \widehat{x}_{t,2} \cdot a_{2,1} + \cdots + \widehat{x}_{t,n} \cdot a_{n,1}$.
      Therefore there needs n^2 multiplications, so the time complexity for this is $O(n^2)$.

   2) $P_{t+1|t} = AP_{t|t}A^T + GQG^T$

      Because $P_{t+1|t}, P_{t|t}, A \in \mathbb{R}^{n\times n}$,
      it needs $O(n^3)$ to multiply matrices. So the time complexity is $O(n^3)$.

   3) $K_{t+1} = P_{t+1|t}C^T(CP_{t+1|t}C^T + R)^{-1}$

      Because $P_{t+1|t} \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{m\times n}$, $K_{t+1} \in \mathbb{R}^{n\times m}$, and $R \in \mathbb{R}^{m\times m}$,

      it needs $O(n^3)$ or $O(m^3)$ for the multiplications, i.e. $max(O(n^3), O(m^3))$.

   4) $\widehat{x}_{t+1|t+1} = \widehat{x}_{t+1|t} + K_{t+1}(y_{t+1} - C\widehat{x}_{t+1|t})$

      This needs $O(n^2)$.

   5) $P_{t+1|t+1} = P_{t+1|t} - K_{t+1}CP_{t+1|t}$

      This needs $O(n^3)$ or $O(m^3)$ for the multiplications, i.e. $max(O(n^3), O(m^3))$.

In overall, the dominant time complexity is related to the matrix multiplication, which needs cubed complexity.
Therefore, the time complexity of the Kalman Filtering is $max(O(n^3), O(m^3))$.

2 Gaussian Hidden Markov Models

1. Given D i.i.d. observations, $\{y_1^{(d)}, \cdots, y_T^{(d)}\}_{d=1}^D$, write down the expected complete log-likelihood of the data.

   Let x_{ti} be

   $$x_{ti} = \begin{cases} 1, & \text{if } x_t = s_i \\ 0, & \text{otherwise} \end{cases}$$

   then we can write the emission probabilities as
   $p(Y_t = y_t | X_t = s_i) = \mathcal{N}(y_t|\mu_i, \Sigma_i) = \prod_{i=1}^N \mathcal{N}(y_t|\mu_i, \Sigma_i)^{x_{ti}}$

   The expected complete log likelihood is defined as $\sum_x p(x|y,\theta) log\, p(x,y|\theta)$.

   $$\begin{aligned} log\, p(x,y|\theta) &= log\{\pi_{x_0} \textstyle\prod_{t=0}^{T-1} a_{x_t,x_{t+1}} \prod_{t=0}^{T} p(y_t|x_t)\} \\ &= log\{\pi_{x_0} \textstyle\prod_{t=0}^{T-1} a_{x_t,x_{t+1}} \prod_{t=0}^{T} \prod_{i=1}^{N} \mathcal{N}(y_t|\mu_i,\Sigma_i)^{x_{ti}}\} \\ &= log\{\textstyle\prod_{i=1}^{N} {\pi_{x_0}}^{x_{0i}} \prod_{t=0}^{T-1} \prod_{i=1}^{N} \prod_{j=1}^{N} {a_{x_t,x_{t+1}}}^{x_{ti}\cdot x_{(t+1)j}} \prod_{t=0}^{T} \prod_{i=1}^{N} \mathcal{N}(y_t|\mu_i,\Sigma_i)^{x_{ti}}\} \\ &= \textstyle\sum_{i=1}^{N} x_{0i} \cdot log \pi_{x_0} + \sum_{t=0}^{T-1} \sum_{i=1}^{N} \sum_{j=1}^{N} x_{ti} \cdot x_{(t+1)j} \cdot log\, a_{x_t,x_{t+1}} \\ &\quad + \textstyle\sum_{t=0}^{T} \sum_{i=1}^{N} x_{ti} \cdot log\, \mathcal{N}(y_t|\mu_i,\Sigma_i)\} \end{aligned}$$

   Let define $\gamma(x_t) = p(x_t|Y)$ and $\xi(x_t, x_{t+1}) = p(x_t, x_{t+1}|Y)$.

   $$\begin{aligned} &\textstyle\sum_x p(x|y,\theta) log\, p(x,y|\theta) \\ &= \textstyle\sum_{i=1}^{N} \gamma(x_{0i}) \cdot log \pi_{x_0} + \sum_{t=0}^{T-1} \sum_{i=1}^{N} \sum_{j=1}^{N} \xi(x_{ti}, x_{(t+1)j}) \cdot log\, a_{x_t,x_{t+1}} \\ &\quad + \textstyle\sum_{t=0}^{T} \sum_{i=1}^{N} \gamma(x_{ti}) \cdot log\, \mathcal{N}(y_t|\mu_i,\Sigma_i)\} \end{aligned}$$

   where $a_{x_t,x_{t+1}}$ is the transition probability from x_t to x_{t+1}.

2. Derive the M-step equations for the EM algorithm for learning the Gaussian HMM model, for the parameters μ_i and Σ_i. You do not need to derive the M-step equations for the other parameters, i.e. π_i and a_{ij}, which are identical to those of the discrete (multinomial) HMM models.

   M-step updates for μ_i and Σ_i are as follows:

   Let $N_i = \sum_{t=0}^{T} \gamma(x_{ti}) = \sum_{t=0}^{T} p(x_t = s_i|Y)$

   $$\begin{aligned} \mu_i^{new} &= \tfrac{1}{N_i} \textstyle\sum_{t=0}^{T} \gamma(x_{ti}) y_t \\ \Sigma_i^{new} &= \tfrac{1}{N_i} \textstyle\sum_{t=0}^{T} \gamma(x_{ti})(y_t - \mu_i^{new})(y_t - \mu_i^{new})^T \end{aligned}$$

3 Conditional Random Fields

The logistic regression model can be written as:

$$p(y|x) = \frac{1}{Z(x)} exp\{\sum_{i=1}^{k} w_i f_i(y, x)\}$$

where w_i is the weight and f_i is the feature function, which is the indicator.

Because $\mathbf{X} = \{X_1, X_2, \cdots, X_k\}$ and $\mathbf{Y} = \{Y\}$, there are k combinations (cliques) of $\mathbf{X}$ and $\mathbf{Y}$, i.e. $\phi_1(X_1, Y), \phi_2(X_2, Y), \cdots, \phi_k(X_k, Y)$.

The definition of a conditional random field can be described as:

$$\begin{aligned} P(y|x) &= \frac{1}{Z(x)} \widetilde{P}(y, x) \\ \widetilde{P}(y, x) &= \prod_{i=1}^{k} \phi_i(D_i) \\ Z(x) &= \sum_y \widetilde{P}(y, x) \end{aligned}$$

When $\phi_i(x_i^l, y^m) = exp\{w_i^{ml} \cdot \mathbb{I}(X_i = x_i^l, Y = y^m)\}$,

$$\begin{aligned} P(y|x) &= \frac{1}{Z(x)} \widetilde{P}(y, x) \\ &= \frac{1}{Z(x)} \widetilde{P}(y, x) \\ &= \frac{1}{Z(x)} \prod_{i=1}^{k} \phi_i(x_i^l, y^m) \\ &= \frac{1}{Z(x)} \prod_{i=1}^{k} exp\{w_i^{ml} \cdot \mathbb{I}(X_i = x_i^l, Y = y^m)\} \\ &= \frac{1}{Z(x)} exp\{\sum_{i=1}^{k} w_i^{ml} \cdot \mathbb{I}(X_i = x_i^l, Y = y^m)\} \end{aligned}$$

which is same with the definition of the logistic regression model.
So, now we can see these are exponential families.

4 An HMM for the Stock Market

By "python hmm.py", the program for 3 questions are executed, producing two graphs (for the question 1 and the question 2) as "q4_1.png" and "q4_2.png", saving the result of the question 3 as "q4_3.txt"

1. Implement the Forward-Backward algorithm and run it using the observations for time points $t = 1, \cdots, 100$. Report the inferred distributions over the hidden states at $t = 1, \cdots, 100$ by plotting the probabilities $P(Z_t = i|X_1, \cdots, X_{100})$ for $i = 1, 2, 3$ over $t = 1, \cdots, 100$. Make sure you label 3 time series (one for each hidden state) in your plot.

   In the plot, the red points are

   $P(Z_1 = 1|X_1, \cdots, X_{100}), P(Z_2 = 1|X_1, \cdots, X_{100}), \cdots, P(Z_{100} = 1|X_1, \cdots, X_{100})$.

   The green ones are the probabilities when $Z_t = 2$, and the blue ones are when $Z_t = 3$.

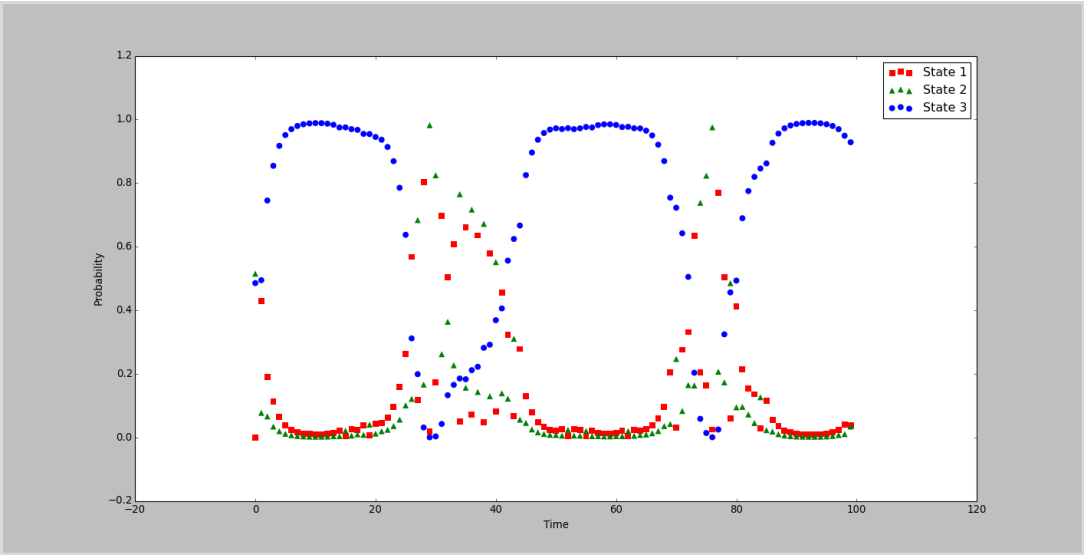

Figure 1: The result of the Forward-Backward algorithm

2. Implement the Viterbi algorithm and find the most likely sequence of hidden states $Z_1, \cdots, Z_{100}$ for the same time period. Report the most likely hidden states over $t = 1, \cdots, 100$ by plotting these values as a time series.

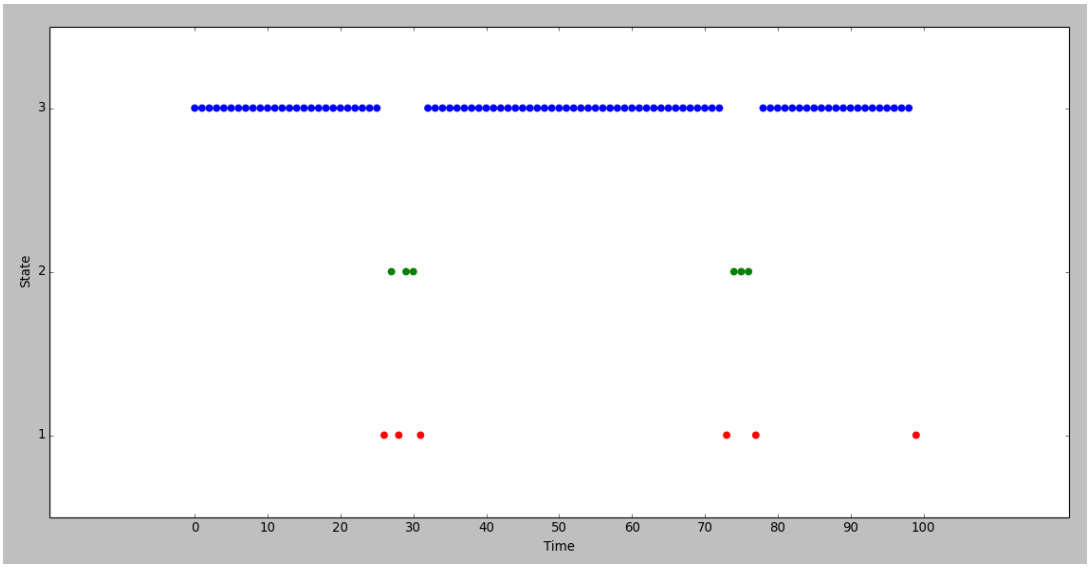

Figure 2: The result of the Viterbi algorithm

3. To make future prediction, predict the output symbols for time points $t = 101, , 128$ by carrying out the following steps for each time point t:

   (a) Run the forward algorithm to estimate $P(Z_{t1}|X_{t101}, \cdots, X_{t1})$. Note that $X_{t101}, \cdots, X_{t1}$ are the ground-truth observations. (Alternatively, you can instead compute $P(Z_t|X_1, \cdots, X_{t1})$, but clarify if you take this approach.)

(b) Compute $P(Z_t)$ from $P(Z_{t1})$ using the transition matrix. Generate the output value X_t by sampling a state z_t from $P(Z|Z_t)$ and then from $P(Z_t|Z_t = z_t)$.

Compare your predictions with the ground truth observations at time points $t = 101, , 128$. What is the percentage of these values that your model predicts correctly? Report the average and variance over 100 runs.

I calculate the prediction using $P(Z_t|X_1, \cdots, X_{t-1})$.

I samples at the unit of sequence from time $t = 101$ to $t = 128$ for 100 times.

The result is,

- Average: 0.707142857143
- Var: 0.00298469387755
- Max: 0.8214285714285714
- Min: 0.5357142857142857

where the average is the mean of the correct probabilities in 100 samples.

In the implementation,
I outputs the posterior probability of Z_t and 100 samples with the correct probability compared with the ground truth.
Finally, I prints the average, the variance, the maximum probability, and the minimum probability.
It shows about 70% correctness with 0.2% variance.