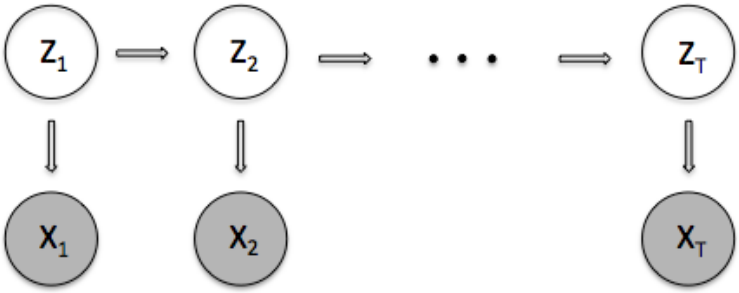
STA561: Probabilistic machine learning

Hidden Markov Models and State Space Models (9/25/13)

Lecturer: Barbara Engelhardt Scribes: Ani Mohan, Lucas Spangher, Shiwen Zhao, Xiaoyang Zhuang

1 Hidden Markov Models



Recall from last lecture:
$Z_{1:T} \equiv Z_1, ..., Z_t, ..., Z_T$ is a series of T *latent* or *hidden random variables*. An example of a latent random variable is an indicator variable for whether or not a woman has preeclampsia at time point t.

When Z_t is a categorical, or multinomial, variable, we will write Z_t as a multinomial vector, or a K-vector with all zeros except for a single one at the kth position. Thus, Z_t^k is a binary indicator for whether or not Z_t is in state $k \in \{1, \ldots, K\}$. That is, the tth hidden state is k iff $Z_t^k = 1$.

$X_{1:T} \equiv X_1, \ldots, X_t, \ldots, X_T$ is a series of T *observed random variables*. An example of an observed random variable at time t is blood pressure or heart rate. While we cannot *see* if a woman has preeclampsia, we can measure her blood pressure, and that information can give us clues as to whether or not she has preeclampsia at time t.

We will estimate the EM algorithm for an HMM. Let $\theta^{(s)}$ be the set of parameter estimates from the s^{th} iteration of the *expectation maximization algorithm*.

The *transition probability* $\mathsf{p}(z_{t+1} \mid z_t)$ is the probability that the model is in state z_{t+1} at time $(t+1)$ given that the model was in state z_t at time t. We will call this transition matrix A and write a_{jk} to describe the probability of $Z_t^k = 1$ conditioned on $Z_{t-1}^j = 1$.

The *emission probabilities* $\mathsf{p}(x_t \mid z_t)$ specify the probability distributions for the data, x_t, given that the model is in state z_t at time t. This distribution can be chosen arbitrarily (e.g., a normal distribution), and will have its own parameters to estimate.

The *data set* will be a set of n trajectories of length T: $\mathcal{D} = \{(x_1, ..., x_T)_1, \ldots, (x_1, \ldots, x_T)_n\}$. Each $X_{t,i}$ is a scalar or a vector, described by the choice of emission distribution.

1.1 Derivation of the E-Step

In the last lecture, we showed that the required expected sufficient statistics for the EM algorithm are $\mathbb{E}[Z_1^k]$ and $\mathbb{E}[Z_t^j Z_{t+1}^k]$. Define $\alpha_t(k) \triangleq \mathsf{p}[Z_t^k = 1 \mid X_{1:t}, \theta^{(s)}]$ and $\beta_t(k) \triangleq \mathsf{p}[X_{t+1:T} \mid Z_t^k = 1, \theta^{(s)}]$.

$$\begin{aligned}
\mathbb{E}[Z_1^k] &\triangleq \mathbb{E}[Z_1^k \mid X_{1:T}, \theta^{(s)}] && (1)\\
&= \mathsf{p}[Z_1^k = 1 \mid X_{1:T}, \theta^{(s)}] && (2)\\
&= \mathsf{p}[Z_1^k = 1 \mid X_{2:T}, X_1, \theta^{(s)}] && (3)\\
&\propto \mathsf{p}[X_{2:T} \mid Z_1^k = 1, X_1, \theta^{(s)}]\mathsf{p}[Z_1^k = 1 \mid X_1, \theta^{(s)}] && (4)\\
&= \mathsf{p}[X_{2:T} \mid Z_1^k = 1, \theta^{(s)}]\mathsf{p}[Z_1^k = 1 \mid X_1, \theta^{(s)}] && (5)\\
&\triangleq \beta_1(k) * \alpha_1(k) && (6)
\end{aligned}$$

Expectation maximization is an iterative procedure that switches between the E-Step and the M-Step, and the computations in the E-Step use information from the observed data $X_{1:T}$ and the parameter estimates $\theta^{(s)}$ from the preceding M-Step. From here we observe that Z_1^k is a Bernoulli random variable, so the expectation is equal to the probability of Z_1^k. The probability function in Eqn. (2) is a **smoothing** function. Eqn. 4 is an application of Bayes Rule. The equality in Eqn. 5 is due to the conditional independence of X_1 and $X_{2:T}$ given that Z_1 has been observed (you can see this by bouncing a Bayes ball on the HMM diagram, after shading Z_1). We are able to write this expectation then as a function of the $\alpha(\cdot)$ and $\beta(\cdot)$ variables.

$$\begin{aligned}
\mathbb{E}[Z_t^j Z_{t+1}^k] &\triangleq \mathbb{E}[Z_t^j Z_{t+1}^k \mid X_{1:T}, \theta^{(s)}]\\
&= \mathsf{p}[Z_t^j Z_{t+1}^k = 1 \mid X_{1:T}, \theta^{(s)}]\\
&= \mathsf{p}[Z_t = j, Z_{t+1} = k \mid X_{1:T}, \theta^{(s)}]\\
&\propto \mathsf{p}[Z_t = j, Z_{t+1} = k, X_{1:T} \mid \theta^{(s)}]\\
&\text{(still conditioning on } \theta^{(s)} \text{)}\\
&\propto \mathsf{p}[X_{t+2:T} \mid Z_{t+1} = k]\mathsf{p}(X_{t+1} \mid Z_{t+1} = k)\mathsf{p}(Z_{t+1} = k \mid Z_t = j)\mathsf{p}(Z_t = j \mid X_{1:t})\\
&= \beta_{t+1}(k) \cdot \mathsf{p}(X_{t+1} \mid Z_{t+1} = k) \cdot a_{jk} \cdot \alpha_t(j) && (7)
\end{aligned}$$

The expectation in the first line simplifies to the posterior probability of observing the transition pair from state j at time t to state k at time $t+1$ when we think of the transition pair as a Bernoulli random variable. From here we use the same tricks to rewrite the transition pair in terms of Z_t and Z_{t+1} as we did for $\mathbb{E}[Z_1^k]$ earlier. The HMM structure allows us to decompose the joint distribution into local computations. For example, $X_{t+2:T}$ is conditionally independent of X_{t+1} given Z_{t+1}, and both are independent of all earlier data in the series, $Z_{1:t}, X_{1:t}$. We are able to write this expectation as a function of our $\alpha_t(\cdot)$ and $\beta_t(\cdot)$ variables.

1.2 Computation of the E-Step

In order to compute our expected sufficient statistics $\mathbb{E}[Z_1^k]$ and $\mathbb{E}[Z_t^j Z_{t+1}^k]$, we must be able to compute $\alpha_t(k)$ and $\beta_t(k)$ (see equations (6) and (7)). In order to do this, we employ the *forward-backward algorithm.*

First, we derive a recursive relation for $\alpha_t(k)$:

$$\begin{aligned}
\alpha_t(k) &\triangleq \mathsf{p}[Z_t^k = 1 \mid X_{1:t}, \theta^{(s)}] \\
&= \mathsf{p}[Z_t^k = 1 \mid X_t, X_{1:t-1}, \theta^{(s)}] \\
&\propto \mathsf{p}[X_t \mid Z_t^k = 1, X_{1:t-1}, \theta^{(s)}]\mathsf{p}[Z_t^k = 1 \mid X_{1:t-1}, \theta^{(s)}] \\
&= \mathsf{p}[X_t \mid Z_t^k = 1, \theta^{(s)}]\mathsf{p}[Z_t^k = 1 \mid X_{1:t-1}, \theta^{(s)}] \\
&= \mathsf{p}[X_t \mid Z_t^k = 1, \theta^{(s)}] \sum_{j=1}^{K} \mathsf{p}[Z_t^k = 1 \mid Z_{t-1}^j = 1, \theta^{(s)}]\mathsf{p}[Z_{t-1}^j = 1 \mid X_{1:t-1}, \theta^{(s)}] \\
&= \mathsf{p}[X_t \mid Z_t^k = 1, \theta^{(s)}] \sum_{j=1}^{K} a_{jk} \cdot \alpha_{t-1}(j) \qquad (8)
\end{aligned}$$

This result is obtained by applying Bayes' rule to the definition of $\alpha_t(k)$ and simplifying according to the conditional independence relations given by the HMM DAG. For example, X_t and $X_{1:t-1}$ are conditionally independent given Z_t (again, you can see this fact by bouncing a Bayes ball on the HMM diagram). To get the recursive relationship, we marginalize Z_{t-1} out of the final equation.

Note that $\alpha_t(k)$ gives the posterior probability that $Z_t^k = 1$, therefore we know that $\sum_{k=1}^{K} \alpha_t(k) = 1$. Once we obtain our estimates for each of the $\alpha_t(k)$ according to equation (8), we then normalize them by dividing by their sum to obtain a proper probability distribution.

Next, we derive a recursive relation for $\beta_t(k)$:

$$\begin{aligned}
\beta_{t-1}(k) &\triangleq \mathsf{p}[X_{t:T} \mid Z_{t-1}^k = 1, \theta^{(s)}] \\
&= \sum_{j=1}^{K} \mathsf{p}[Z_t^j = 1, X_t, X_{t+1:T} \mid Z_{t-1}^k = 1, \theta^{(s)}] \\
&= \sum_{j=1}^{K} \mathsf{p}[X_{t+1:T} \mid X_t, Z_t^j = 1, Z_{t-1}^k = 1, \theta^{(s)}]\mathsf{p}[X_t \mid Z_t = j, Z_{t-1}^k = 1, \theta^{(s)}]\mathsf{p}[Z_t^j = 1 \mid Z_{t-1} = k, \theta^{(s)}] \\
&= \sum_{j=1}^{K} \mathsf{p}[X_{t+1:T} \mid Z_t^j = 1, \theta^{(t)}]\mathsf{p}[X_t \mid Z_t^j = 1, \theta^{(s)}]\mathsf{p}[Z_t^j = 1 \mid Z_{t-1}^k = 1, \theta^{(t)}] \\
&= \sum_{j=1}^{K} \beta_t(j)\mathsf{p}[X_t \mid Z_t^j = 1, \theta^{(s)}]a_{kj} \qquad (9)
\end{aligned}$$

The reasoning here is essentially the same as the reasoning for $\alpha_t(k)$, however this time we are moving backwards through the DAG so we marginalize over Z_t to obtain the previous $\beta_{t-1}(k)$. Note that $\beta_t(k)$ represents the probability of the future data given knowledge of the latent state at time point t. Unlike the $\alpha_t(k)$, there is no particular reason why the $\beta_t(k)$ should sum to one over all the values for Z_{t-1}.

forward-backward algorithm:

We will now describe how we can compute the α and β vectors that we will use to compute the Expected Sufficient Statistics for the EM algorithm. Let $\pi_k \triangleq \mathsf{p}[Z_1^k = 1|\theta^{(s)}]$ denote the probability distribution of the initial state.

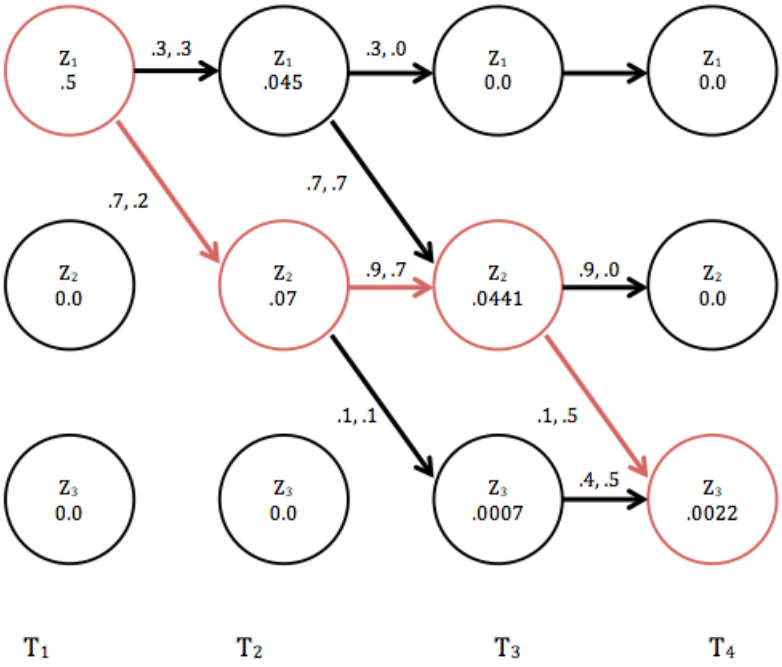
forward pass:
Starting from $\pi_{1:k}$ and $\alpha_1(k) \propto \mathsf{p}[X_t \mid Z_t^k = 1, \theta^{(s)}]\pi_k$, use Equation (8) to compute $\alpha_2(k)$ up to $\alpha_T(k)$.

backward pass:
Starting at time T, we incorporate each 'future' observation into our estimate for time t using equation (9).

We compute $\beta_{T-1}(k)$, $\beta_{T-2}(k)$, and so forth until we compute $\beta_1(k)$. Note that we need $\beta_T(k)$ to start this process. Recall that $\beta_T(k)$ gives the probability of seeing the future data at time T, but we have not collected any future data yet. We therefore begin this backward recursion at $\beta_T(k) = 1$.

The necessary emission and transition probabilities for both the forward and backward pass are given by $\theta^{(s)}$. It is important to observe that the *forward-backward algorithm* is a relatively efficient algorithm for calculating our expected sufficient statistics with a runtime of $\mathsf{O}(K^2T)$, linear in the time points T. With a limited number of latent states K, we can thus perform EM for HMMs that model very long sequences quickly. This is all due to the Markov assumption and the joint probability factorization properties that conditional independencies enable.

The *Viterbi algorithm* produces the most likely sequence of hidden states, and is a canonical example of *dynamic programming*. It is equivalent to a procedure for the MAP estimate, and therefore uses the maximum product instead of the sum product. That is, the sums over K in the above equation for β are changed to max over K (with some details glossed over). This algorithm has a runtime that is linear in the number of time points, $\mathsf{O}(K^2T)$.



This is a representation of the goal of the Viterbi algorithm. This demonstrates four steps in time, in which the maximum over the transition probabilities are calculated for each state. The latent position shifts at each point in time to maximize the probability of the the sequence of latent states.

1.3 Computation of the M-Step

To complete the M step, we need to take derivative of $Q(\theta, \theta^{(i-1)})$ with respect to the parameters in $\theta = (A, \pi, \eta)$. Recall from last lecture the definition of the expected complete log likelihood for the HMM:

$$Q(\theta, \theta^{(i-1)}) = \sum_{k=1}^{K} \mathbb{E}[Z_1^k \mid X_{1:T}, \theta^{(i-1)}] \log \pi_k \tag{10a}$$

$$+ \sum_{t=1}^{T-1} \sum_{j,k=1}^{K} \mathbb{E}[Z_t^j Z_{t+1}^k \mid X_{1:T}, \theta^{(i-1)}] \log a_{jk} \tag{10b}$$

$$+ \sum_{t=1}^{T} \mathbb{E}[\log \mathsf{p}(X_t \mid Z_t, \eta)]. \tag{10c}$$

Updating π, the initial state vector:

We must maximize $Q(\theta, \theta^{(i-1)})$ over π subject to the constraint that $\sum_{k=1}^{K} \pi_k = 1$. This is left as an exercise, and can be derived using Lagrange multipliers, constraining $\sum_{k=1}^{K} \pi_k = 1$.

$$\pi^{\text{new}} = \frac{1}{n} \sum_{k=1}^{K} \mathbb{E}[Z_1^k \mid X_{1:T}, \theta^{(i-1)}]$$

Observe that due to the interpretation of Z_1^k, this is simply the counts of "soft assignment". That is, the update for π is the posterior expected number of time points that are assigned to state k at time 1.

Updating A, the transition matrix:

This is similar to the update for π because $Q(\theta, \theta^{(i-1)})$ has the same functional form for specific entries of both A and π. This can be seen by comparing equations (10a) and (10b) for π_k and a_{jk} respectively. Furthermore, just like π, the matrix A must be constrained to be stochastic as it represents transition probabilities of the underlying Markov chain on the latent states $Z_{1:T}$. Thus, updating matrix A is equivalent to counting the number of times (in posterior expectation) the latent state j at time $t-1$ transitioned to the latent state k at time t:

$$a_{ij}^{\text{new}} = \frac{\sum_{t=2}^{T} \mathbb{E}(z_{t-1}^j z_t^k \mid X_{1:T}, \theta^{(s-1)})}{\sum_{t=2}^{T} \sum_j \mathbb{E}(z_{t-1}^j z_t^k \mid X_{1:T}, \theta^{(s-1)})}$$

Updating η, the parameters of the emission probability distribution:

Since η only appears on line (10c) of $Q(\theta, \theta^{(i-1)})$, we can ignore the other parts of the equation for the purposes of finding the update for η. Beyond this, the parameter updates depend on the definition of $\mathsf{p}(X_t \mid Z_t, \eta)$.

2 State Space Models

These models differ from Hidden Markov Models in that the latent variables are continuous. They are the same, though, in that they model a discrete collection of time points, with an underlying Markov chain that we assume has a stationary distribution.

As with the Hidden Markov Model, we again have

- a *transition model* for the underlying Markov chain $Z_t \sim g(U_t, Z_{t-1}, \epsilon_t)$
- an *observation model* that specifies the observed data distribution conditional on the latent state, $Y_t = h(Z_t, U_t, \delta_t)$

- here, U_t is the control or *input signal*.

For now, our main goal is to calculate the probability of hidden states given observations and signals (*filtering*; Equation (11)), but *prediction* of future states and observations is also an important goal for many research applications. The probability of the hidden states is given by:

$$p(Z_t \mid Y_{1:T}, U_{1:T}, \theta) \tag{11}$$

For now, we will drop the U_t terms and assume $g(\cdot)$ and $h(\cdot)$ are both linear Gaussian.

$$Z_t = A_t Z_{t-1} + B_t U_t + \epsilon_t \tag{12a}$$
$$Y_t = C_t Z_t + D_t U_t + \delta_t, \tag{12b}$$

where $\epsilon_t \sim \mathcal{N}(0, Q_t)$ and $\delta_t \sim \mathcal{N}(0, R_t)$. The parameters of the full state space model are $\Theta = \{A_t, B_t, C_t, D_t, Q_t, R_t\}$. With an assumption of stationarity, there is no need to keep the subscript of t in Θ.

This model is often referred to as a *Linear Dynamical System*, and the above modeling choices support *exact computation* of relevant posterior quantities. Computation is straightforward using the *Kalman filter*. If the initial latent state distribution is Gaussian, then subsequent states will also be. Just as for the HMM, exact computation will be linear in the number of time points.

2.1 Online parameter learning

Recursively perform least squares regression by defining $\mu_t = A_t\mu_{t-1} + K_t(Y_t - C_t A_t \mu_{t-1})$ where $K_t = \Sigma_t C_t^T R_t^{-1}$ is known as the *Kalman Gain matrix*. The parameter updates are then given by

$$\hat{\theta}_t = \hat{\theta}_{t-1} + \frac{1}{\sigma^2}\Sigma_{t|t}(Y_t - C_t Z_{t-1})C_t$$

The recursive updates for $\mu_\cdot$ and $\Sigma_\cdot$ are then given by

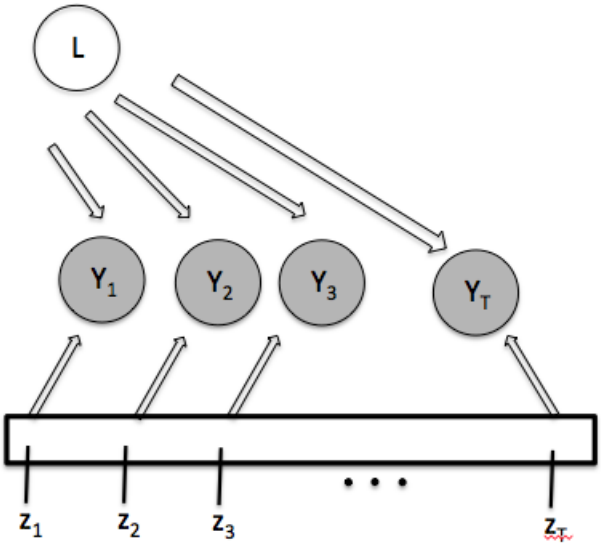
$$\mu_{t|t-1} \triangleq A_t\mu_{t-1} + B_t U_t$$
$$\Sigma_{t|t-1} \triangleq A_t\Sigma_{t-1}A_t^T + Q_t$$

Putting it all together, we obtain the probability distribution of our latent state at time t

$$\mathsf{p}(Z_t \mid Y_{1:t-1}, U_{1:t}) = \mathcal{N}(Z_t \mid \mu_{t|t-1}, \Sigma_{t|t-1}).$$

2.2 Motion Tracking Application

Consider the rectangular box here as representing a continuous 1-dimensional space being traversed by a robot. The robot must identify its own position (latent states Z) using its sensor measurements, Y, which fluctuate according to the relative position of the robot to ambient landmarks, L. See the figure below for a graphical representation of this SSM.


L
Y1
Y2
Y3
YT
z1
z2
z3
zT