MSA101/MVE187 2021 Lecture 8 Hidden Markov Models and state space models. Kalman filters.

Petter Mostad

Chalmers University

September 21, 2021

Overview

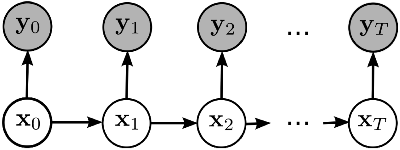
- The Bayesian paradigm: Define Y_{data}, Y_{pred}, and a stochastic model. Make predictions for Y_{pred} by first finding (or generating a sample from) the posterior for a model parameter vector θ.
- We have looked at example models where θ consists of a handful of parameters.
- Today we turn to models with with a "time-structure": At each time point, the structure of the stochastic model is the same, but variables change over time.

Time-structured models

- Many examples of sequential data: The results for a sports team, data from a self-driving car, data from speach analysis, ...
- The structuring variable need not be time: Another example is DNA sequences.
- We assume "data of the same format" are observed at time points t. A possible goal: Predict data at future times.
- *Continuous time models*: Possible, but not treated here.
- We assume data $y_0, y_1, \ldots, y_T$ observed at times $0, 1, \ldots, T$.
- Models can get complicated because of complicated dependencies between the y_i.
- A powerful way to formulate a model: *Assume there is a sequence of* hidden variables $x_0, x_1, \ldots, x_T$ *so that* x_i *stores all information relevant to predict* $y_i, y_{i+1}, \ldots, y_T$.

State space models

- We assume there is a *Markov chain* of hidden variables $x_0, x_1, \ldots, x_T$ that can be used to predict the observed variables $y_0, \ldots, y_T$:

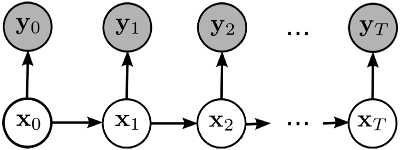


- Note that the distribution of y_i is modelled only in terms of x_i.
- Note that $x_0, \ldots, x_T$ is a Markov chain, so that, for example

$$\pi(x_{i+1} \mid x_0, x_1, \ldots, x_i) = \pi(x_{i+1} \mid x_i).$$

- Many statements of conditional independencies can be read off the graph of dependencies above, for example: Given the value of x_i, the variables $y_i, \ldots, y_T$ are indepenent of variables $y_1, \ldots, y_{i-1}$.
- We will only consider *homogeneous* Markov chains: The variables x_i are of the same type, and the conditional distributions $\pi(x_i \mid x_{i-1})$ are all the same.
- We will also assume that the *emission distributions* $\pi(y_i \mid x_i)$ are the same for all i (and the variables y_i are of the same type).

State space models



- Thus, to specify such a state space model we need to specify

$$\pi(x_0) \qquad \pi(x_i \mid x_{i-1}) \qquad \pi(y_i \mid x_i)$$

- There is a possibility to model also direct dependencies between y_i and y_{i+1}, but as y_i and y_{i+1} are generally observed, adjusting the theory is easy, and not considered here.
- The random variables x_i and y_i may be of any type, and may be vectors!
- When x_i are discrete variables with a finite number of possible values, we call the above a *Hidden Markov Model* (HMM).
- If the variables are all (multivariate) normal, and if the dependencies $\pi(x_i \mid x_{i-1})$ and $\pi(y_i \mid x_i)$ are linear, we call the above a *linear dynamical system*.

Toy example

In this lecture we will work with a simple toy example of an HMM:

- The hidden variables $x_1, \dots, x_N$ have possible values $1, \dots, M$, and transition probabilities in the chain are (initially):

$$x_i \text{ given } x_{i-1} \text{ is } \begin{cases} \text{with prob. } 1/3\text{:} & x_{i-1}+1 \text{ if possible, otherwise } x_{i-1}. \\ \text{with prob. } 1/3\text{:} & x_{i-1}. \\ \text{with prob. } 1/3\text{:} & x_{i-1}-1 \text{ if possible, otherwise } x_{i-1}. \end{cases}$$

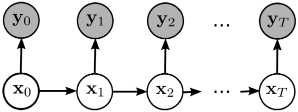
- The observed variables y_i are Poisson distributed with expectations given by the x_i:
- **Presentation break for R simulations**

Inference for state space models

- Within the Bayesian paradigm, one might want to find the full posterior
$$\pi(x_0, \dots, x_T \mid y_0, \dots, y_T).$$
Usually represented by sample sequences $x_0, \dots, x_T$.
- An easier goal is to find the marginal posterior for each x_i:
$$\pi(x_i \mid y_0, \dots, y_T)$$
This will be our main focus.
- In fact, our algorithm will be a special case of a more general algorithm (called, e.g., "message passing" or "sum-product algorithm"). We hope to return to this.
- Assume we have an HMM, so that the x_i have a finite set of possible values. A goal might be to find the sequence $x_0, \dots, x_T$ of values such that
$$\pi(x_0, \dots, x_T \mid y_0, \dots, y_T)$$
is maximized. We look at the Viterbi algorithm for this below.
- The distributions $\pi(x_i \mid x_{i-1})$ and $\pi(y_i \mid x_i)$ might have unknown parameters. We may return to making inference also for such parameters.

The Forward-Backward algorithm

Message passing applied to a *Hidden Markov Model*.



Objective: Compute the marginal posterior distribution of every x_i given data $y_0, \dots, y_T$: Use $\pi(x_i \mid y_0 \dots, y_T) \propto_{x_i} \pi(y_{i+1}, \dots, y_T \mid x_i)\pi(x_i \mid y_0, \dots, y_i)$ and

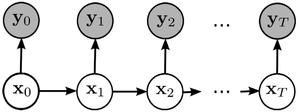
1. Forward: For $i = 0, \dots, T$ compute $\pi(x_i \mid y_0, \dots, y_i)$ using

$$\begin{aligned} \pi(x_i \mid y_0, \dots, y_i) &\propto_{x_i} \pi(y_i \mid x_i)\pi(x_i \mid y_0, \dots, y_{i-1}) \\ &= \pi(y_i \mid x_i) \int \pi(x_i \mid x_{i-1})\pi(x_{i-1} \mid y_0, \dots, y_{i-1})\, dx_{i-1} \end{aligned}$$

2. Backward: For $i = T-1, \dots, 0$ compute $\pi(y_{i+1}, \dots, y_T \mid x_i)$ using

$$\pi(y_{i+1}, \dots, y_T \mid x_i) = \int \pi(y_{i+2}, \dots, y_T \mid x_{i+1})\pi(y_{i+1} \mid x_{i+1})\pi(x_{i+1} \mid x_i)\, dx_{i+1}$$

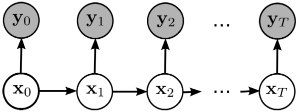
The Forward-Backward algorithm for our HMM example



- The hidden chain $x_0 \to \cdots \to x_N$ is a random walk on the integers $\{1, \ldots, M\}$.
- The (prior) transition probabilities from x_i to x_{i+1} is to increase with 1 (if possible) with probability $1/3$, to decrease with 1 (if possible) with probability $1/3$, and otherwise stay put.
- We use the model $y_i \mid x_i \sim \text{Poisson}(x_i)$ and assume the y_i are observed.
- We use the Forward-Backward algorithm to find the marginal posterior probability for each x_i.
- **Presentation break for R computations**

The Viterbi algorithm

We consider an HMM where the x_i have a finite state space $\{1, \ldots, M\}$:



Objective: Compute the vector $x_0, \ldots, x_T$ which maximizes the posterior $\pi(x_0, \ldots, x_T \mid y_0, \ldots, y_T)$, i.e., maximizes $\pi(x_0, \ldots, x_T, y_0, \ldots, y_T)$.

- First formulation of an algorithm: Sequentially, for $i = 0, \ldots, T$, compute and store
  - For each $j = 1, \ldots, M$, the sequence $\hat{x}_0, \ldots, \hat{x}_i$ maximizing $\pi(\hat{x}_0, \ldots, \hat{x}_i, y_0, \ldots, y_i)$ while $\hat{x}_i = j$.
  - For each $j = 1, \ldots, M$, the value of the maximum above.
- Note that

  $$\pi(x_0, \ldots, x_i, y_0, \ldots, y_i) = \pi(x_0, \ldots, x_{i-1}, y_0, \ldots, y_{i-1}) \cdot \pi(x_i \mid x_{i-1}) \pi(y_i \mid x_i)$$

  Thus the results for stage i with $\hat{x}_i = j$ can be found by finding the $\hat{x}_{i-1}$ in $\{1, \ldots, M\}$ maximizing

  $$\pi(\hat{x}_0, \ldots, \hat{x}_{i-1}, y_0, \ldots, y_{i-1}) \cdot \pi(x_i = j \mid \hat{x}_{i-1})$$

The Viterbi algorithm

- Thus results for the i'th step in the sequence can be computed by considering all combinations of values for x_i and x_{i-1} together with results from the $i-1$'th step.
- Improved and final formulation of the algorithm: For each i and j, you only need to store $\hat{x}_{i-1}$, not the whole sequence $\hat{x}_0, \ldots, \hat{x}_{i-1}, \hat{x}_i = j$. THEN: At any point, $(\hat{x}_1, \ldots, \hat{x}_i)$ can be reconstructed tracing backwards through stored information.
- **Presentation break for computations in R**

Kalman filters

- The Forward-Backward algorithm applied to the case where all variables are (multivariate) normal and all dependencies are linear is called the *Kalman filter*.
- Because of the Normal-Normal conjugacy, all the distributions we compute in the Forward-Backward algorithm become Normal distributions.
- Specifically, assume in the multivariate case

$$\begin{aligned}
\pi(x_i \mid x_{i-1}) &= \text{Normal}\left(x_i; Ax_{i-1} + b, P^{-1}\right) \\
\pi(y_i \mid x_i) &= \text{Normal}\left(y_i; Cx_i + d, Q^{-1}\right) \\
\pi(x_0) &= \text{Normal}\left(x_0; \mu_0, R^{-1}\right)
\end{aligned}$$

  Then
  - the Forward algorithm produces a recursive formula for the parameters of the normal distribution $\pi(x_i \mid y_0, \dots, y_i)$,
  - the Backward algorithm produces a recursive formula for the parameters of a normal distribution proportional to $\pi(y_{i+1}, \dots, y_T \mid x_i)$,
  - The normal-normal conjugacy produces from this parameters for the normal distribution $\pi(x_i \mid y_0, \dots, y_T)$.

Formulas for a simple 1D Kalman filter

To simplify formulas we look at the 1D example

$$\begin{aligned}
\pi(x_i \mid x_{i-1}) &= \text{Normal}\left(x_i; x_{i-1}, \tau_1^{-1}\right) \\
\pi(y_i \mid x_i) &= \text{Normal}\left(y_i; x_i, \tau_2^{-1}\right) \\
\pi(x_0) &= \text{Normal}\left(x_0; \mu_0, \tau_0^{-1}\right)
\end{aligned}$$

- For $i = 0, \dots, T$, we define values a_i, α_i such that

  $$\pi(x_i \mid y_0, \dots, y_i) = \text{Normal}(x_i; a_i, \alpha_i^{-1})$$

  and use the Forward algorithm to obtain a recursive formula.
- For $i = T-1, \dots, 0$, we define values b_i, β_i such that

  $$\pi(y_{i+1}, \dots, y_T \mid x_i) \propto_{x_i} \text{Normal}(x_i; b_i, \beta_i^{-1})$$

  and use the Backward algorith to obtain a recursive formula.
- The normal-normal conjugacy gives directly that

  $$\pi(x_i \mid y_0, \dots, y_T) = \text{Normal}\left(x_i; \frac{\alpha_i a_i + \beta_i b_i}{\alpha_i + \beta_i}, (\alpha_i + \beta_i)^{-1}\right)$$

Forward recursive formula

- For $i = 0$ we get

$$\begin{aligned}
\pi(x_0 \mid y_0) \quad &\propto_{x_0} \quad \text{Normal}(y_0; x_0, \tau_2^{-1})\,\text{Normal}(x_0; \mu_0, \tau_0^{-1}) \\
&\propto_{x_0} \quad \text{Normal}\left(x_0; \frac{\mu_0\tau_0 + y_0\tau_2}{\tau_0 + \tau_2}, (\tau_0 + \tau 2)^{-1}\right)
\end{aligned}$$

so $a_0 = \frac{\mu_0\tau_0 + y_0\tau_2}{\tau_0+\tau_2}$ and $\alpha_0 = \tau_0 + \tau_2$.

- For $i = 1, \dots, T$,

$$\begin{aligned}
&\pi(x_i \mid y_0, \dots, y_i) \\
\propto_{x_i} \quad & \pi(y_i \mid x_i) \int \pi(x_i \mid x_{i-1})\pi(x_{i-1} \mid y_0, \dots, y_{i-1})\, dx_{i-1} \\
= \quad & \text{Normal}(y_i; x_i, \tau_2^{-1}) \int \text{Normal}(x_i; x_{i-1}, \tau_1^{-1})\,\text{Normal}(x_{i-1}; a_{i-1}, \alpha_{i-1}^{-1})\, dx_{i-1} \\
= \quad & \text{Normal}(y_i; x_i, \tau_2^{-1})\,\text{Normal}(x_i; a_{i-1}, \tau_1^{-1} + \alpha_{i-1}^{-1}) \\
\propto_{x_i} \quad & \text{Normal}\left(x_i; \frac{(\tau_1^{-1} + \alpha_{i-1}^{-1})^{-1} a_{i-1} + \tau_2 y_i}{(\tau_1^{-1} + \alpha_{i-1}^{-1})^{-1} + \tau_2}, ((\tau_1^{-1} + \alpha_{i-1}^{-1})^{-1} + \tau_2)^{-1}\right)
\end{aligned}$$

so $a_i = \frac{(\tau_1^{-1}+\alpha_{i-1}^{-1})^{-1} a_{i-1} + \tau_2 y_i}{(\tau_1^{-1}+\alpha_{i-1}^{-1})^{-1} + \tau_2}$ and $\alpha_i = (\tau_1^{-1} + \alpha_{i-1}^{-1})^{-1} + \tau_2$.

Backward recursive formula

- Set $b_T = 0$, $\beta_T = 0$.
- For $i = T-1, \ldots, 0$, we get

$$
\begin{aligned}
& \pi(y_{i+1}, \ldots, y_T \mid x_i) \\
= \quad & \int \pi(y_{i+2}, \ldots, y_T \mid x_{i+1}) \pi(y_{i+1} \mid x_{i+1}) \pi(x_{i+1} \mid x_i)\, dx_{i+1} \\
\propto_{x_i} \quad & \int \text{Normal}(x_{i+1}; b_{i+1}, \beta_{i+1}^{-1}) \cdot \text{Normal}(y_{i+1}; x_{i+1}, \tau_2^{-1}) \cdot \\
& \text{Normal}(x_{i+1}; x_i, \tau_1^{-1})\, dx_{i+1} \\
\propto_{x_i} \quad & \int \text{Normal}\left(x_{i+1}; \frac{\beta_{i+1} b_{i+1} + \tau_2 y_{i+1}}{\beta_{i+1} + \tau_2}, (\beta_{i+1} + \tau_2)^{-1}\right) \cdot \\
& \text{Normal}(x_i; x_{i+1}, \tau_1^{-1})\, dx_{i+1} \\
= \quad & \text{Normal}\left(x_i; \frac{\beta_{i+1} b_{i+1} + \tau_2 y_{i+1}}{\beta_{i+1} + \tau_2}, (\beta_{i+1} + \tau_2)^{-1} + \tau_1^{-1}\right)
\end{aligned}
$$

  so $b_i = \frac{\beta_{i+1} b_{i+1} + \tau_2 y_{i+1}}{\beta_{i+1} + \tau_2}$ and $\beta_i = ((\beta_{i+1} + \tau_2)^{-1} + \tau_1^{-1})^{-1}$.
- **Presentation break for computations in R**