Probabilistic Time Series Analysis: Lab 5

Tim Kunisky

September 26, 2018

The Discrete Hidden Markov Model

Same setting as with LDS, only now all variables are discrete:

$$\begin{array}{ccccccccc} \cdots & \rightarrow & z_{t-2} & \rightarrow & z_{t-1} & \rightarrow & z_t & \rightarrow & \cdots \\ & & \downarrow & & \downarrow & & \downarrow & & \\ \cdots & & x_{t-2} & & x_{t-1} & & x_t & & \cdots \end{array}$$

States in $\{1, \ldots, K\}$; observations in $\{1, \ldots, H\}$.

$$\mathbb{P}(z_t = \ell \mid z_{t-1} = k) = A_{k\ell} \quad (\boldsymbol{A} \in \mathbb{R}^{K \times K} \text{ is the } \textit{transition matrix})$$

$$\mathbb{P}(z_1 = k) = \pi_k \quad (\boldsymbol{\pi} \text{ is the } \textit{initial distribution})$$

$$\mathbb{P}(x_t = i \mid z_t = k) = B_{ki} \quad (\boldsymbol{B} \in \mathbb{R}^{K \times H} \text{ is the } \textit{emission matrix})$$

The model parameters are $\boldsymbol{\theta} = \{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi}\}$.

Exercise: Let's write down the joint distribution $\mathbb{P}(\boldsymbol{z}, \boldsymbol{x} \mid \boldsymbol{\theta})$.

Computation 1: Likelihood Function

The likelihood is

$$\mathbb{P}(\boldsymbol{x} \mid \boldsymbol{\theta}) = \sum_{k_1=1}^{K} \cdots \sum_{k_T=1}^{K} \mathbb{P}(z_1 = k_1, \ldots, z_T = k_T, \boldsymbol{x} \mid \boldsymbol{\theta}).$$

How do we compute it without summing over K^T terms? We use the Markov property of the z_t, which allows a recursion:

$$\begin{aligned} \alpha_t(k) &= \mathbb{P}(x_1, \ldots, x_t, z_t = k). \\ &= \sum_{\ell=1}^{K} \mathbb{P}(x_{[1:t-1]}, z_{t-1} = \ell)\mathbb{P}(z_t = k \mid z_{t-1} = \ell)\mathbb{P}(x_t \mid z_t = k) \\ &= \sum_{\ell=1}^{K} \alpha_{t-1}(\ell) A_{\ell,k} B_{k,x_t}. \end{aligned}$$

At the end we can take $\mathbb{P}(\boldsymbol{x} \mid \boldsymbol{\theta}) = \sum_{k=1}^{K} \alpha_T(k)$.

Likelihood Function with Linear Algebra

The recursion for the likelihood is:

$$\alpha_t(k) = \sum_{\ell=1}^{K} \alpha_{t-1}(\ell) A_{\ell,k} B_{k,x_t}.$$

This is simple to express in terms of matrix operations: let's say $\boldsymbol{\alpha}_t \in \mathbb{R}^K$, then

$$\begin{aligned}
\boldsymbol{\alpha}_t^\top &= \boldsymbol{\alpha}_{t-1}^\top A \operatorname{diag}(\boldsymbol{B}_{\bullet,x_t}) \\
&= \boldsymbol{\alpha}_1^\top A \operatorname{diag}(\boldsymbol{B}_{\bullet,x_t}) A \operatorname{diag}(\boldsymbol{B}_{\bullet,x_{t-1}}) \cdots A \operatorname{diag}(\boldsymbol{B}_{\bullet,x_2})
\end{aligned}$$

Every entry of each $\operatorname{diag}(\boldsymbol{B}_{\bullet,x_t})$ is at most 1 (it is a probability!) and $\|A\| \leq 1$, so these quantities will shrink quickly—hence the normalization step from lecture.

Computation 2: MAP Estimation / Decoding

Now, we see $\boldsymbol{x}$ and want to compute

$$\hat{\boldsymbol{z}} = \text{optimizer in } \max_{k_1} \cdots \max_{k_T} \mathbb{P}(z_1 = k_1, \ldots, z_T = k_T, \boldsymbol{x} \mid \boldsymbol{\theta}).$$

Formally, this is just like the last problem, but with "sum" → "max." So, we can use the same recursion:

$$\begin{aligned}
v_t(\ell) &= \max_{k_1,\ldots,k_{t-1}} \mathbb{P}(x_{[1:t]}, z_{[1:t-1]} = k_{[1:t-1]}, z_t = \ell) \\
&= \max_{k_{t-1}} \max_{k_1,\ldots,k_{t-2}} \mathbb{P}(x_{[1:t-1]}, z_{[1:t-2]} = k_{[1:t-2]}, z_{t-1} = k_{t-1}) \\
&\qquad \mathbb{P}(z_t = \ell \mid z_{t-1} = k_{t-1}) \mathbb{P}(x_t \mid z_t = \ell) \\
&= \max_{k_{t-1}} v_{t-1}(k_{t-1}) A_{k_{t-1},\ell} B_{\ell,x_t}.
\end{aligned}$$

However: we want to actually learn $\hat{\boldsymbol{z}}$, so we also keep track of the optimizers k_i as we go along.

Coding Assignment

- Find `labs/lab5/lab5-student.ipynb` on the course Github. (It will be easier if you clone the whole repository.)

- We're going to do very real data science! The task is **part of speech (POS) tagging**.
- This tags English words by their grammatical function in a sentence (nouns, verbs, adjectives, adverbs, conjunctions, etc.), which is not always easy: "I will **book** the hotel" vs. "I will read the **book**."
- You will implement HMM for this on the standard Wall Street Journal dataset, containing 39,815 sentences for training and 1,700 sentences for testing.