12-752: Data-Driven Building Energy Management
Assignment#4

Due: Sunday December 10th by 11:59pm on Canvas

December 4, 2017

You should implement your work in a Jupyter Notebook and it should contain proper documentation and be suitable to serve as a professional report (e.g., please be diligent in creating professional looking graphs including axis labels and units, as well as writing in a clear and technical language).

You will submit a .ipynb file, as well as a PDF version of the same file, through Canvas. Your notebook (and the PDF) should containing all of your solutions and comments to the tasks below.

1 Hidden Markov Models

During our review of Paper #3 we learned about Hidden Markov Models (HMMs) and their factorial variant, both in the context of Non-Intrusive Load Monitoring. In this assignment we are going to use a simple toy example to implement some of the basic ideas of HMMs using baby steps.

To start, let's assume that you want to model the aggregate electricity consumption of a household that contains only 2 electrical appliances. Since we are only dealing with 2 appliances, we can collapse the Factorial HMM into an HMM that has one state for every combination of appliance states (i.e., we can have a single latent variable Z that maintains the state of both appliances together). Thus, if we assume each of the individual appliances has two operating states, the resulting HMM will have a total of 4 hidden states, i.e. $Z \in \{(\text{off}, \text{off}), (\text{on}, \text{off}), (\text{off}, \text{on}), (\text{on}, \text{on})\}$. For notational short-hand we will use: $(0,0), (0,1), (1,0), (1,1)$ to refer to the states.

We will denote the latent variables by Z and the observations by X, thus for the joint distribution of the HMMs the following holds:

$p(x, z) = p(z_0) \prod_{t=1}^{N} p(x_t|z_t) p(z_t|z_{t-1})$

Let's further assume that the emission probabilities ($p(x_t|z|t)$) are Gaussian with a state-dependent mean and standard deviation, i.e.:

$p(x_t|z_t) = \mathcal{N}(x_t|\mu_{z_t}, \sigma^2_{z_t})$

> **Task #1 [10%]:** *If following this procedure you were to create an HMM for 22 two-state appliances, how much memory would your computer need if you wanted to fit the state-transition probability matrix into main memory with 4 byte precision (i.e., each value in the transition matrix would require 4 bytes of memory)?*

1.1 Sampling from an HMM

For the first portion of our implementation, we assume knowledge of the intial state, $\pi = p(z_0)$, state transition, $A = p(z_t|z_{t-1})$, and emission probabilities, $B = p(x_t|z_t)$ (i.e., we have all the parameters $\lambda = \{\pi, A, B\}$ for the HMM model already specified). Given these quantities, your task is to sample a time series of possible power values and hidden states, i.e. sample from $p(x, z)$. The easiest way to do this is to first sample a sequence of latent states z by first sampling z_0 and then successively sample from $p(z_t|z_{t-1})$. Given the sequence of z, you can then sample an x_t for every z_t.

> **Task #2 [20%]:** *Sample $(x_t, z_t) \sim p(x, z)$ with a length of 100, i.e., $t \in 1, 2, \ldots, 100$*

> **Task #3 [10%]:** *Compute the likelihood of your sample under the specified model (i.e, under the given λ)*

> **Task #4 [5%]:** *Plot the sequence of observed power values x*

1.2 Inference with an HMM

For the next set of tasks, we are interested in inference. In other words, given a sequence of observations x, we want to tell something about the states of appliances.

First, we need to create a matrix $P1 \in (0, 1)^{4\times 100}$ where every entry contains the $p(z_t|x_t)$. You can compute this quantity from $p(x_t|z_t)$, i.e.: $p(z_t|x_t) = \frac{p(x_t|z_t)p(z_t)}{\sum_z p(x_t|z_t=z)p(z_t=z)}$

If we furthermore assume that $p(z_t)$ is uniform, then:
$p(z_t|x_t) = \frac{p(x_t|z_t)}{\sum_z p(x_t|z_t=z)}$

> **Task #5 [10%]:** *Generate matrix $P1$ as indicated above*

Now let's see if we can improve on our previous results by incorporating temporal dependencies between latent variables.

For this, we can estimate $p(z_t|x_{1:t})$, i.e. the probability of being in a latent state z_t given all previous observations $x_{1:t} = x_1, ..., x_t$.

We can derive this as follows:

$$
\begin{aligned}
p(z_t, x_{1:t}) &= \sum_{z_{t-1}} p(z_t, z_{t-1}, x_{1:t}) && (1)\\
&= \sum_{z_{t-1}} p(x_t|z_t, z_{t-1}, x_{1:t-1})p(z_t|z_{t-1}, x_{1:t-1})p(z_{t-1}, x_{1:t-1}) && (2)\\
&= p(x_t|z_t) \sum_{z_{t-1}} p(z_t|z_{t-1})p(z_{t-1}, x_{1:t-1}) && (3)
\end{aligned}
$$

Note that the left-hand side (LHS) of the equation is the same as the last term of the right-hand side (RHS) when decrementing t by 1. Thus, we can compute $p(z_t, x_{1:t})$ recursively, i.e. first compute $p(z_1, x_1)$, followed by $p(z_2, x_{1:2})$, ..., until $p(z_t, x_{1:t})$ following the equation above.

> **Task #6 [20%]:** *Compute $p(z_t, x_{1:t})$ for all t and store the values in a matrix $P2$*

Task #7 [10%]: *Normalize the probabilities, i.e. compute $p(z_t|x_{1:t})$ and store the values in a matrix P3*

Task #8 [10%]: *Plot P1 alongside P3 (use* `plt.imshow(Px, aspect='auto', interpolation='nearest')` *and compare to the ground truth.*

Task #9 [5%]: *Did incorporating temporal dependencies help?*