Assignment 3
Statistical Methods in Applied Computer Science
DD2447

Jim Holmström 890503-7571
jimho@kth.se

December 16, 2012

In all cases you are supposed to provide as sound algorithms of the types described in the course. Summing over exponentially many terms and other solutions that always gives exponential time, will not be considered sufficient.

Exercise 1/2 MAP switch setting and position on model railway
Imagine a model railway with a single train. You know the map of the tracks including the position of all the switches, but you don't know current states of the switches, or where the train is currently located. Each switch has three connections: $\{0, L, R\}$. If the train comes from the direction of L or R, it always leaves in the direction 0. If the train comes from the direction 0, it will leave in either direction L or R, depending on the state of the switch. The switch has prior probability 1/2 for each direction, but will remain the same throughout the train run. You are receiving a stream of signals from the train, each signal specifying the direction in which train has passed a switch: $\{0L, 0R, L0, R0\}$; you do not know, however, which switch the train has passed. Also, the sensors are noisy, and with a certain probability p, the train reports a random signal instead of a real direction in which it passed the switch.

(1a) Given a sequence of switch signals $\{y_i\}_{i=1}^{T}$ and the settings of the switches compute MAP position z_t.

(1b) Given a sequence of switch signals $\{y_i\}_{i=1}^{T}$ and the corresponding positions $\{z_i\}_{i=1}^{T}$ compute MAP switch settings.

(2) Given the map of the railroad and a sequence of switch signals $\{y_i\}_{i=1}^{T}$. Your task is to compute MAP switch and current position.

Solution. Represent the connections on switch α with $\{\alpha_0, \alpha_L, \alpha_R\}$.

The outline for this solution is to reduce the original problem to one which fulfills the Markov assumption and can thus be used without being a approximation in a ordinary HMM.

"Current position" can either be the position of the last passed switch which is hinted by the assignment having a list of positions for the switches given or it could be defined as being in between two connections. If we have the first case we simply pick the switch from the last of the two connections we are between according to MAP solution.

Let $\cdot \leftrightarrow \cdot$ denote a undirected link [1] between connections in the given map.

Construct a switch-graph [2] G of the map where each switch is represented by a triangle with the side LR in bold, [3] each connection is a node, and the edges is given by the map information $\cdot \leftrightarrow \cdot$.

A simple example of a map would be the two switches α, β with the links $\{\alpha_L \leftrightarrow \beta_L, \alpha_0 \leftrightarrow \beta_R, \alpha_R \leftrightarrow \beta_0\}$

If we were to use this representation of states we stumble into problems when you are in state α_0 since you don't know if you came from α_L or α_R and thus cannot decide the correct distribution for the observation. A possible solution [4] to this would be to perform a directed lifting inside the switches of the graph G and the links becomes undirected edges between the new states, denote this by G'. So $\forall \alpha \in$ switches we have the 4 states: $\{\alpha_0 \to \alpha_L, \alpha_0 \leftarrow \alpha_L, \alpha_0 \to \alpha_R, \alpha_0 \leftarrow \alpha_R\}$. Which gives us a total of $4T$ states for G'.

Note that G' is sparse with only $O(T)$ edges both in the undirected inherited from G as well as the lifted directed ones.

Each link from the original map constructs a constraint on the transition matrix A. And can be listed as follows; the sparsity renders most transitions 0 along with the diagonal and thus we will only list the non-zero ones. Looking locally at a connection we have either it being a in- or output and the same goes for the connected neighbour to it. In this case in- and output basically means input$= \{\alpha_0\}$ and output$= \{\alpha_L, \alpha_R\}$.

For output linked to output we have:

$$\begin{aligned} \alpha_L \leftrightarrow \beta_L \Rightarrow (\alpha_0 \leftarrow \alpha_L) &\to (\beta_0 \to \beta_L) = 1, \\ (\beta_0 \leftarrow \beta_L) &\to (\alpha_0 \to \alpha_L) = 1 \end{aligned} \tag{1}$$

[1]Use the word "link" instead of "connection" to avoid name collision

[2]Not a real graph yet because of the switches, they could be deduced at this step but it will just make it harder.

[3]Arbitrary representation of a switch but it catches that it is one object and that you cannot pass between L and R.

[4]Another possible solution is to lift it on the links and then lift that graph again, which was my first try. I think that also worked but it became a real mess.

For input to output (and output to input by symmetry) we have:

$$\begin{aligned}\alpha_0 \leftrightarrow \beta_L \Rightarrow (\alpha_0 \leftarrow \alpha_L) &\rightarrow (\beta_0 \leftarrow \beta_L) = 1,\\ (\alpha_0 \leftarrow \alpha_R) &\rightarrow (\beta_0 \leftarrow \beta_L) = 1,\\ (\beta_0 \leftarrow \beta_L) &\rightarrow (\alpha_0 \rightarrow \alpha_L) = P_\alpha,\\ (\beta_0 \leftarrow \beta_L) &\rightarrow (\alpha_0 \rightarrow \alpha_R) = 1 - P_\alpha\end{aligned} \tag{2}$$

For input to input we have:

$$\begin{aligned}\alpha_0 \leftrightarrow \beta_0 \Rightarrow (\alpha_0 \leftarrow \alpha_L) &\rightarrow (\beta_0 \rightarrow \beta_L) = P_\beta,\\ (\alpha_0 \leftarrow \alpha_L) &\rightarrow (\beta_0 \rightarrow \beta_R) = 1 - P_\beta,\\ (\alpha_0 \leftarrow \alpha_R) &\rightarrow (\beta_0 \rightarrow \beta_L) = P_\beta,\\ (\alpha_0 \leftarrow \alpha_R) &\rightarrow (\beta_0 \rightarrow \beta_R) = 1 - P_\beta,\end{aligned} \tag{3}$$

And finally from this all possible local cases comes from symmetry.

The observation matrix (maps state to observation probability) can be constructed by simply taking

$$P(y|\alpha_i \rightarrow \alpha_j) = \begin{cases} 1 - \frac{3p}{4} & ij = y \\ p/4 & ij \neq y \end{cases} \tag{4}$$

Note that all the information about which one of the switches that signaled is lost. And of course the representation of a row in transition matrix and the observation matrix most agree to have a sensible matrix multiplication.

For the first problem (1a) we set P_α to either 0 or 1 corresponding to the switch settings and then we simply get the most probable hidden state sequence with Viterbi which is known to be MAP. We then trivially post-process the hidden state sequence to extract the position in the wanted form, for example state $(\alpha_i \rightarrow \alpha_j) \leftrightarrow \alpha$.

For the second problem (1b) it becomes harder since a exact inference of the transition probability in a general HMM is intractable. We do need to use the sparseness of parameters in our transition matrix to make it tractable or we need to show that approximative algorithms in fact returns the exact solution for our case. The intractability comes from that we have to sum up over all possible hidden state sequences to calculate the probability for some things. For example it is not tractable to check the probability for each combination of switch settings most probable hidden state sequence (where the state sequence can be either given or not).

There exist some papers on utilizing the sparsity in the model, like " Constrained Hidden Markov Models" - Sam Roweis, but they are sparse. And my hunch is reducing the number of free parameters from T^2 to $O(T)$ in the transition matrix will not make exact inference to be sub-exponential.

The conclusion on this is that the solution is probably to show that the local optimization solution is also global and thereby being exact. The probability for a local solution to be a global one is greatly increased when going from a complete transition matrix to a $O(n)$. One could probably show that the optimization problem is convex for this type of sparse matrix while general transition matrix is not.

To solve (2) we first infer the switch settings like above and then use the found transition matrix to find the most probable hidden state sequence with Viterbi.

Exercise 3 Gibbs sampler for posterior of magic word generative model
The following generative model generates K sequences of length N: $\{s_i\}_{i=1}^{K}$ where $s_i = \{s_{i,j}\}_{j=1}^{N}$. All sequences are over the alphabet $[M]$. Each of these sequences has a "magic" word of length w hidden in it and the rest of the sequence is called background.

First $\forall i$, a start position r_i for the magic word is sampled uniformly from $[N - w + 1]$. Then the j:th positions in the words are sampled from $q_j(x)$, which is $Cat\,(x|\theta_j)$ where θ_j has a $Dir\,(\theta_j|\alpha)$ prior. All other positions in the sequences are sampled from the background distribution $q(x)$, which is $Cat\,(x|\theta)$ where θ has a $Dir\,(\theta|\alpha')$ prior.

Solution.