KTH Computer Science
and Communication

Implementation of “The Train” problem

Statistical Methods in Applied Computer Science, DD2447
Final Project

JIM HOLMSTRÖM JIMHO@KTH.SE

Teacher: Jens Lagergren

Contents

Chapter 1

The Problem

1.1 Problem formulation [1]

G is undirected and all vertices has degree 3. At each vertex the edges are labeled 0,L, and R (an edge can have different labels at different vertices). So a vertex is a switch. Start positions and switch settings have uniform priors.

A switch setting is a function $\sigma : V(G) \to \{L, R\}$, which has the natural interpretation. By a position we mean a pair (v, e), where $v \in V(G)$ and $e \in E(G)$, with the interpretation that the train has passed v and exited through e.

(We leave out G and a instead have G implicit for all distributions and expressions.)

Below we give a DP algorithm for $p(s, O|\sigma)$. We will estimate $p(\sigma|O)$ using MCMC and then $p(s|O)$ using:

$$p(s|O) = \sum_{\sigma} p(s, \sigma|O) = \sum_{\sigma} p(s|\sigma, O)p(\sigma|O) = \sum_{\sigma} \frac{p(s, O|\sigma)p(\sigma|O)}{p(O|\sigma)} \tag{1.1}$$

The probability p is 0.05. Given $G, \sigma, s \in V(G)$ (s is a stop position), $O \in \{L, R, 0\}^T$ (observed switch signals), we can compute $p(s, O|\sigma)$ using dynamic programming (DP) and in each step compute the probability of going from some position s' to s in t steps and observing $O_{1:t} = \{o_i\}_{i=1}^t$ when the switch settings are σ. By doing this for all stop positions s and then summing out the stop position, we obtain $p(O|\sigma)$ in time $\mathcal{O}(N^2T)$, where $N = |V(G)|$.

The states in our HMM are positions. The transition probabilities are always 1, i.e., given how we enter a vertex it is uniquely determined how we exit (since switches are fixed). Also, when passing a switch the correct direction of the label of the position is emitted with probability $1 - p$ and any different direction is emitted with probability $p/2$.

[1] This formulation has some minor differences but it should still be equivalent with the original one.

Let $c(s,t)$ be computed as below (we want $c(s,t)$ to be the probability of going from some position s' to $s = (v,e)$ in t steps and observing $O_{1:t}$). Let $f = (u,v)$ and $g = (w,v)$ be the two edges that are incident with v but different from e.

$$c(s,t) = \begin{cases} 1/N, & t = 0 \\ \left[c((u,f),t-1) + c((w,g),t-1)\right](1-p) & e = 0 \wedge o_t = 0 \\ \left[c((u,f),t-1) + c((w,g),t-1)\right]p & e = 0 \wedge o_t \neq 0 \\ \left[c((u,f),t-1)\right](1-p) & e = L \wedge o_t = L \wedge f = 0 \\ \left[c((u,f),t-1)\right]p & e = L \wedge o_t \neq L \wedge f = 0 \\ \left[c((u,f),t-1)\right](1-p) & e = R \wedge o_t = R \wedge f = 0 \\ \left[c((u,f),t-1)\right]p & e = R \wedge o_t \neq R \wedge f = 0 \end{cases} \tag{1.2}$$

Chapter 2

Methodology

2.1 Derivation

The underlying model for this problem is a HMM but to calculate $p(s|O)$ we need to reform it to the form of a HMM.

2.1.1 Marginalization $p(s|O) = \sum_{\sigma} p(s, \sigma|O)$

The first marginalization in (??) is to get fixed σ and thereby fixed transition probabilities inside the $\sum \sigma$ to make the sub-problem tractable, the downside to this is that we get $|\{L, R\}^N| = \mathcal{O}\left(2^N\right)$ summations which is intractable. This will later be approximated to overcome the intractability with a Markov chain Monte Carlo (MCMC) method.

2.1.2 Conditional probability $p(s, \sigma|O) = p(s|\sigma, O)p(\sigma|O)$

From the definition of conditional probability we can factor out the $\sigma|O$ distribution and by this setting it up for the MCMC later on.

2.1.3 Conditional probabilty $p(s|\sigma, O) = \frac{p(s,O|\sigma)}{p(O|\sigma)}$

To get it $p(s, O|\sigma)$ to match the form of $\bar{\alpha}_t = p(O_{1:T}, \bar{s}|model)$ with the fw and we can still calculate $p(O|\sigma)$ by marginalization on s.

2.2 HMM

c vs c_{σ}

2.3 MCMC

*basic basic MCMC abstract theory *MCMC used for estimating $Ef(\sigma)$ without the intractible process of going through all σ. *why is $Ef(\sigma) = \text{sum } f(\sigma)p(\sigma) = \sum_{sampled} f(\sigma)/N$ *how it relates to our problem *$s\bar{i}gma$ graph with the weights $Q(\bar{\sigma}, \bar{\sigma}')$ which have 2^N states and $(2^N)^2$ connections note about being to vast to search through (dim-devil) *different algorithms for implementing MCMC *compare the different ones shortly(what they assume or need for example) *explain why we chose it *how we chose Q what are the requirements for it *hamming distance measure and picking close ones (as we should accordingly) *why hamming? a more natural distance for switch settings than say euclidean or such. *step size $(\delta) = 1$ (tuning parameter) *abstractly relate to the graph (what is this type of graph called?) *describe the marginalization over states in $p(\ sigma|o)$ and why we need to do it

Chapter 3

Results & Discussion

*MCMC *convergence analysis *trace plot *autocorrelation? *test different distributions for step size. *step-size effect *to small step size => autocorrelation high (slow convergence) *to big step size => low acceptence rate