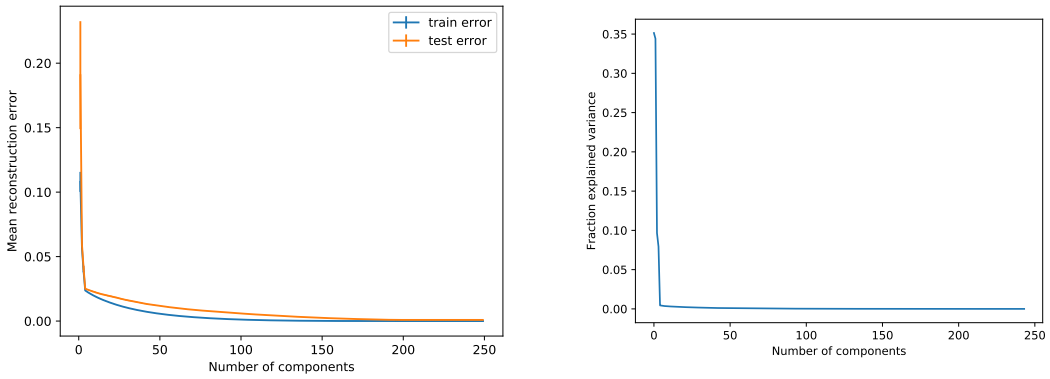
COMP 652: ASSIGNMENT 3

CARLOS G. OLIVER (260425853)

1. Q1: PCA



(A) (B)

FIGURE 1

2. Q2: SPECTRAL METHODS FOR WEIGHTED AUTOMATA

2.1. **(a).** We use a property of Hankel matrices which is: if $\text{rank}(H_f) = n \Rightarrow$ there exists a weighted finite automaton $\mathcal{A}$ with n states such that $g = g_{\mathcal{A}}$ for some function $g : \Sigma^* \to \mathbb{R}$ where Σ^* is the set of all strings that can be generated from alphabet Σ. In this example, the function g counts the number of 1s in a string generated from $\Sigma = \{0, 1\}$ the Hankel matrix H_g would take the form:

Date: April 14, 2017.

$$H_g = \begin{array}{c} \\ \lambda \\ 0 \\ 1 \\ 11 \\ 111 \\ 110 \\ \vdots \end{array} \begin{array}{c} \begin{array}{cccc} \lambda & 0 & 1 & 11 \end{array} \quad \ldots \\ \begin{bmatrix} 0 & 0 & 1 & 2 & \ldots \\ 0 & 0 & 1 & 2 & \\ 1 & 1 & 2 & 3 & \\ 2 & 2 & 3 & 4 & \\ 3 & 3 & 4 & 5 & \\ 2 & 2 & 3 & 4 & \\ \vdots & & & & \ddots \end{bmatrix} \end{array}$$

The above matrix is arranged so all possible strings composed only of 1s, $Q = \{1\}^N \quad \forall N \in \mathbb{Z}^+$ precede all other strings $(\Sigma^* - Q)$ with the exception of $\{0\}^1$ and λ. If we order prefixes and suffixes in S in increasing N we can easily see that any row m in this sub block H_g^1 of H_g is a linear combination in the form $H_g^1(m,\cdot) = 2H_g^1(m,\cdot) - H_g^1(m-1,\cdot)$. The remaining entries in H_g are formed by prefixes and suffixes containing 0s which do not contribute to the evaluation of g and therefore can also be obtained from rows or columns in H^1. Therefore the rank of H_g is 2 where only the λ or $\{0\}^1$ and the $\{1\}^1$ contribute to the row rank. The same can be shown for the column rank.

2.2. **(b).** If f is a probability distribution over Σ^* then we have $\sum_{s\in\Sigma^*} f(s) = 1$ gives a probability for every string in Σ^*. Then $f_{sub}(w)$ is the probability that the string w occurs in any word generated by Σ^* since u and v represent all possible prefixes and suffixes to the word w.

2.3. **(c).** Let $\Sigma_{w\in\Sigma^*} A_w \equiv S$.

$$\begin{aligned} S = (I - A_0 - A_1)^{-1} &= (I - A_0 - A_1)S = I \\ (I - A_0 - A_1)SS^{-1} &= S^{-1} \\ (I - A_0 - A_1)^{-1} &= S \end{aligned} \tag{1}$$

Now we use this identity to compute the sum of the function f over all words as $\Sigma_{w\in\Sigma^*} f(w)$. We can express the function over a string as

$$f(w) = \alpha_0^T A w \alpha_\infty \tag{2}$$

So for all the words we have

$$\begin{aligned} \sum_{w\in\Sigma^*} f(w) &= \sum_{w\in\Sigma^*} \alpha_0^T A_w \alpha_\infty \\ &= \alpha_0^T \Big(\sum_{w\in\Sigma^*} A_w \Big) \alpha_\infty \\ &= \alpha_0^T (I - A_0 - A_1)^{-1} \alpha_\infty \end{aligned} \tag{3}$$

We can use this property to compute $f_{subsrting}(w)$ using an automaton by recognizing that the contribution of transition matrices from u and v to the sum can be reduced to the same identity as above as they sum over all of Σ^*.

$$
\begin{aligned}
f_{substring}(w) &= \sum_{u\in\Sigma^*, v\in\Sigma^*} f(uwv) \\
&= \sum_{u\in\Sigma^*, v\in\Sigma^*} \alpha_0^T A_u A_w A_v \alpha_\infty \\
&= \alpha_0^T (I - A_0 - A_1)^{-1} A_w (I - A_0 - A_1)^{-1} \alpha_\infty
\end{aligned} \tag{4}
$$

2.4. **(d).** Because the word uwv is in Σ^* and the training sample S is drawn from f also over Σ^*, having an automaton that represents this function would also give us a way to learn the function $f_{substring}$. From S we can construct the empirical Hankel matrix $\hat{H}_{f_{substring}}$ and recover the relevant sub-blocks. We can then perform SVD on H and solve for the necessary transition weights A_w and initial/final vecctors α_0, α_∞ from which we can produce the weighted automaton $\hat{A}$. In order to recover an estimate of f we would have to sample from the automaton over all $w \in \Sigma^*$ as this would compute the probability distribution over all words in Σ^*.

3. Q3: Method of moments and multiview model

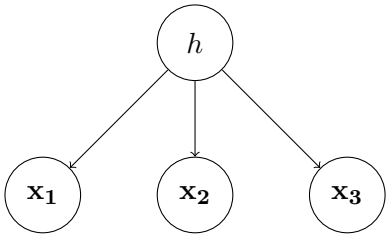


FIGURE 2. Graphical representation of conditional independence relation between random variables $\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}$ and h.

Since the $\mathbf{x_t}$ are conditionally independent given h we can decompose the conditional expectation of the cross moments.

$$
\begin{aligned}
\mathbb{E}[\mathbf{x_1} \otimes \mathbf{x_2} | h = j] &= \mathbb{E}[\mathbf{x_1} | h = j] \otimes \mathbb{E}[\mathbf{x_2} | h = j] && \text{by conditional independence} \\
&= \mu_{t=1,j} \otimes \mu_{t=2,j} \in \mathbb{R}^{d_1 \times d_2} && \text{by question statement} \\
&= \sum_{i=1}^{k} w_i \mu_{1,i} \otimes \mu_{2,i}
\end{aligned} \tag{5}
$$

$$\mathbb{E}[\mathbf{x_1} \otimes \mathbf{x_2} \otimes \mathbf{x_3} | h = j] = \sum_{i=1}^{k} w_i \mu_{1,i} \otimes \mu_{2,i} \otimes \mu_{3,i} \in \mathbb{R}^{d_1 \times d_2 \times d_3} \tag{6}$$

Because the tensor products of the $\mathbf{x_t}$ give rise to non-symmetric tensors, we cannot apply the standard tensor method of moments as eigenvalue decomposition requires symmetric matrices [1]. Other techniques can be employed but are beyond the scope of this question.

4. Q4: Coupled Hidden Markov Models

4.1. **(a).** Parametrization of CHMM:

- Initial probabilities:

$$\begin{aligned} P(s_0 = s) \quad \forall s \in S \\ P(u_0 = u) \quad \forall u \in U \end{aligned} \tag{7}$$

- Transition Probabilities

$$\begin{aligned} P(s_{t+1} = s' | s_{t-1} = s, u_{t-1} = u) \quad \forall s', s, u \in S \cup U \\ P(u_{t+1} = u' | u_{t-1} = u, s_{t-1} = s) \quad \forall u', u, s \in S \cup U \end{aligned} \tag{8}$$

- Emission Probabilities

$$\begin{aligned} P(y_t = y | s_t = s) \quad \forall s \in S, y \in Y \\ P(z_t = y | u_t = u) \quad \forall u \in U, z \in Z \end{aligned} \tag{9}$$

4.2. **(b).** We want to compute $P((y_0, z_0), (y_1, z_1), ...(y_T, z_T))$. We can obtain this probability by marginalizing over the states in the joint over the full model.

(10)

$$\begin{aligned} &P((y_0, z_0), \ldots, (y_T, z_T)) \\ &= \sum_{s_0,\ldots,s_T,u_0,\ldots u_T} P((y_0, z_0), \ldots (y_T, z_T), s_0, \ldots, s_T, u_0, \ldots, u_T) \\ &= \sum_{s_0,\ldots,s_T,u_0,\ldots u_T} P(s_0)P(u_0) \prod_{t=1}^{T} P(s_t | s_{t-1}, u_{t-1}) \prod_{t=1}^{T} P(u_t | s_{t-1}, u_{t-1}) \prod_{t=0}^{T} P(y_t | s_T) \prod_{t=0}^{T} P(z_t | u_t) \\ &= \sum_{s_T} P(y_T | s_T) \sum_{u_T} P(z_T | u_T) \sum_{s_0,\ldots,s_{T-1},u_0,\ldots u_{T-1}} P(s_T | s_{T-1}) P(u_T | u_{t-1}) P(s_0) P(u_0) \\ &\quad \prod_{t=1}^{T-1} P(s_t | s_{t-1}, u_{t-1}) \prod_{t=1}^{T-1} P(u_t | s_{t-1}, u_{t-1}) \end{aligned}$$

We obtain the second step simply by factoring the graph into its independent components according to the graphical model. By rearranging terms we see that the full joint distribution can be written as a product of the distribution at the final time T and the

joint over all times $T-1$ which suggests a dynamic programming algorithm which we can write in the following form:

(11)
$$P((y_0,z_0),...(y_T,z_T)) = \sum_{s_T,u_T} P((y_0,z_0),(y_1,z_1),...(y_T,z_T),u_T,s_T)$$
$$= \sum_{s_T,u_T} P(z_T|u_T)P(y_T|s_T) \sum_{S_{T-1}} P(s_T|s_{T-1})P(u_T|u_{T-1})P((y_0,z_0),...(y_T,z_T),s_{T-1},u_{T-1})$$

We define a matrix $A_t(s,u)$ of dimensions $|S|$ and $|U|$ that will store the probability of the observation sequence until time t and of state s_t and u_t. In the uncoupled version of HMM we have a vector instead of a matrix. The entries of this matrix have the following form:

$$A_t(s,u) = P((y_0,z_0),...(y_t,z_t),s_t=s,u_t=u) \tag{12}$$

At $t=0$ we have our base case which is a function of the initial probabilities s_0 and u_0:

(13)
$$A_0(s,u) = P(y_0,z_0,u_0=u,s_0=s) = P(s_0=s)p(u_0=u)P(y_0|s_0=s)P(z_0|s_0=s) \qquad \forall s,u \in S,U$$

For all subsequent time steps $t=1,...t=T$ we can use the recursive expression derived in (10):

$$A_t(s,u) = P(y_t|s_t)P(z_t|u_t) \sum_{s_{t-1},u_{t-1}} P(s_t|s_{t-1},u_{t-1})P(u_t|s_{t-1},u_{t-1})A_{t-1}(s,t) \tag{14}$$

Rewriting this in more compact form:

$$P(y_0,z_0),...(y_t,z_t)) = \sum_{s_T,u_T} A_T(s,u) \tag{15}$$

4.3. **(c).** This task can be accomplished with a modified version of the Viterbi algorithm which is a dynamic programming technique.

We wish to compute the set of hidden states with the maximum probability given the observed sequence:

$$\underset{s_0u_0,s_1u_1,...,s_Tu_T}{\operatorname{argmax}} P(s_0u_0,s_1u_1,...,s_Tu_T|(y_0,z_0),...(y_t,z_t)) \tag{16}$$

Using simple rules of probability we can rewrite the conditional probability:

$$\underset{s_0u_0,s_1u_1,...,s_Tu_T}{\operatorname{argmax}} \frac{P(s_0u_0,s_1u_1,...,s_Tu_T,(y_0,z_0),...(y_t,z_t))}{P((y_0,z_0),...(y_t,z_t))} \tag{17}$$

We note that the denominator is just the probability of the observation sequence and does not depend on the hidden states so we can omit it from further calculations.

Now we can adapt the Viterbi algorithm (as described on Wikipedia) as follows:

To simplify notation we let:

$$P(u_t = u'|u_{t-1} = u) \equiv \mathbb{U}_{u,u'} \tag{18}$$
$$P(s_t = s'|s_{t-1} = s) \equiv \mathbb{S}_{s,s'}$$

Now we consider the observed sequence $y_0z_0, y_1z_1, ..., y_t, z_t$. Let the probability of the *most likely* state sequence $s_0u_0, s_1u_1, ...s_iu_j$ ending at states $i \in S$ and $j \in U$ be stored in the 3D matrix V where

$$V_{0,i,j} = P(y_0|i)P(z_0|j)P(s_0)P(u_0) \tag{19}$$

Is the sequence at $t = 0$ so we just have the emission probabilities multiplied by the initial state probabilities. For subsequent time steps we have the same as above, instead now we include transition probability factors instead of initial state probabilities, as well as the probability of the sequence of states which led to time t which is just $V_{t-1,s,u}$:

$$V_{t,i,j} = \underset{s\in S,u\in U}{\operatorname{argmax}} P(y_t|i)P(z_t|j)\mathbb{S}_{s,i}\mathbb{U}_{u,j}V_{t-1,s,u} \tag{20}$$

This will result in a dynamic programming table storing the probability of all possible paths through the CHMM. Instead of a 2D table as in the standard Viterbi algorithm we will have a 3D table of dimensions $|T| \times |S| \times |U|$. To retrieve the most likely sequence of hidden states, we store back pointers at every V_t that lead to the state that maximized V_{t-1}.

4.4. **(d).** If chains are coupled every k time steps, then for time steps that are multiples of k we have the same form as the CHMM and for those that are not multiples of k we have two independent chains. This would allow us to decompose the probabilities for time steps in parts (b) and (c) into those that are multiples of k i.e. $\{t \in \pi_k \quad \text{if} \quad t \mod k = 0\}$ and thus remain as before, and those that are not multiples of k, i.e. $t \notin \pi_k$.

5. Notes

Collaboration with Navin Mordani and Jeremy Georges-Filtreau.

References

[1] Animashree Anandkumar, Rong Ge, Daniel J Hsu, Sham M Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research*, 15(1):2773–2832, 2014.