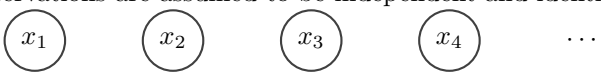
1 Subjects

- Basic algorithms and applications
- Building models and selecting model parameters

2 Notes

2.1 Motivation for HMM

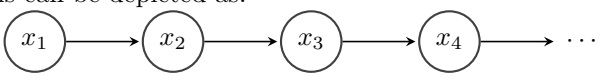
Our predictions are based on models of observed data, a simple model is that observations are assumed to be independent and identically distributed.



But this assumption is not always the best, an example is measurements of weather patterns, daily value of stocks etc. In these cases, a model that shows how each observation depends on the previous observations might be better.

$$p(x_n|x_1,\ldots.x_{n-1}) = p(x_n|x_{n-1})$$

This can be depicted as:

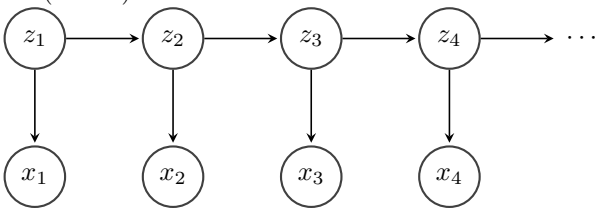


This chain of observations is a 1st-order Markov chain. The joint probability of observing some sequence of N observations is then:

$$p(x_1,\ldots,x_N) = \prod_{n=1}^{N} p(x_n|x_1,\ldots,x_{n-1}) = p(x_1)\prod_{n=2}^{N} p(x_n|x_{n-1})$$

For example, suppose the weather has two states, sun and rain. It might be that if the weather is sunny, then with probability $\frac{6}{7}$ it will stay sunny and $\frac{1}{7}$ it will start to rain however if it's already raining then the probability that it will keep raining is not $\frac{1}{7}$ but instead it might be $\frac{2}{3}$, with just $\frac{1}{3}$ chance it will turn sunny. So the probability of some observation, depends on the last observation, or "what state are we currently in".

Now suppose the state that influences the weather is not necessarily what the last observation was, but rather some other state that is hidden to us? For example wether there is a high or low pressure. If the hidden variables are discrete and form a Markov chain, then we can model it as a hidden Markov model (HMM).



The joint distribution is then:

$$p(x_1, \ldots, x_N, z_1, \ldots, z_N) = p(z_1) \left[\prod_{n=2}^{N} p(z_n|z_{n-1}) \right] \prod_{n=1}^{N} p(x_n|z_n)$$

Then here we can see that $p(z_1) \left[\prod_{n=2}^{N} p(z_n|z_{n-1}) \right]$ models the probability that we are in state z_n given states $z_1, \ldots, z_{n-1}$. And the probability $\prod_{n=1}^{N} p(x_n|z_n)$ is the probability that, given we are in some state z_n, we make the observation x_n.

2.1.1 Modeling transmission probabilities

Notation: we write the hidden variables as positional vectors, i.e. if $z_n = (0, 0, 1)$ then the model in step n is in state $k = 3$.

Given that the hidden variables had to be discrete in order to model them with a HMM, we know that the have K states. Since we know that the amount of states are limited, we can model the transmission probabilities as a $K \times K$ table called A. And we can describe the initial state given by the probability distribution $p(z_1)$ as a K vector π. We can describe this formally as:

$$A_{jk} = p(z_{nk} = 1|z_{n-1,j} = 1), \quad \pi_k = p(z_{1k} = 1)$$

We then have that:

$$\sum_k A_{jk} = 1, \quad \sum_k \pi_k = 1$$

We can then write the math cleverly like this:

$$p(z_n|z_{n-1}, A) = \prod_{k=1}^{K} \prod_{j=1}^{K} A_{jk}^{z_{n-1,j} z_{nk}}, \quad p(z_1|\pi) = \prod_{k=1}^{K} \pi_k^{z_{1k}}$$

Now, this may look confusing. but remember that the hidden variables are positional vectors. That is:

$$z_1 = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

So when we say $\pi_k^{z_{1k}}$ then z_{1k} will be equal to 0 for all entries except one where it is 1. Thus we will have a product of $\pi_0^0 \cdots \pi_{i-1}^0 \cdot \pi_i^1 \cdot \pi_{i+1}^0 \cdots \pi_K^0$. So it is really just a way of extracting a single value out of π and A.

2.1.2 Modeling emission probabilities

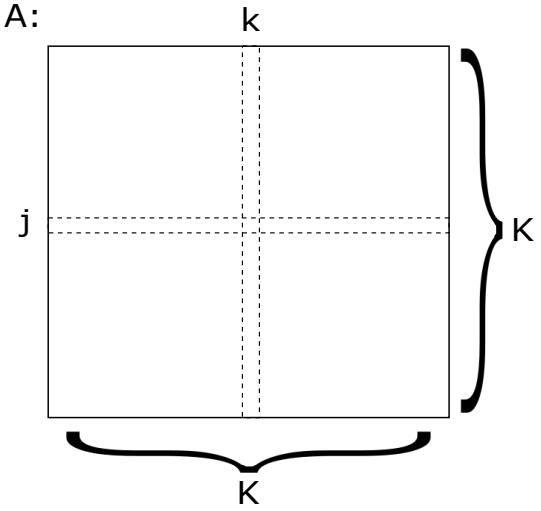
Similar to the transmission probabilities, if we assume that the observed values x_n are discrete (e.g. D different symbols), then the emission probabilities ϕ can be modeled as a $K \times D$ table of probabilities, which for each state K specifies the probability of observing some symbol.

$$p(x_n|z_n, \phi) = \prod_{k=1}^{K} p(x_n|\phi_k)^{z_{nk}} = \prod_{k=1}^{K} \prod_{d=1}^{D} \phi_{kd}^{x_{nd}, z_{nk}}$$

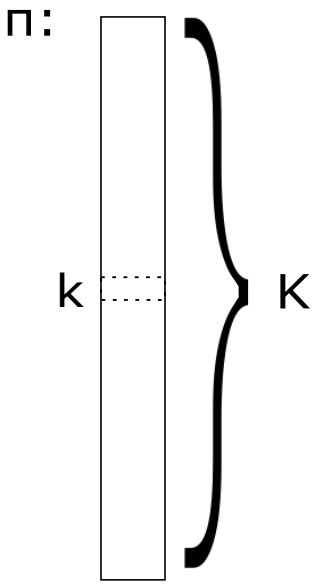
We then get the joint probability distribution of the HMM:

$$p(X, Z|\Theta) = p(z_1|\pi) \left[\prod_{n=2}^{N} p(z_n|z_{n-1}, A) \right] \prod_{n=1}^{N} p(x_n|z_n, \phi)$$

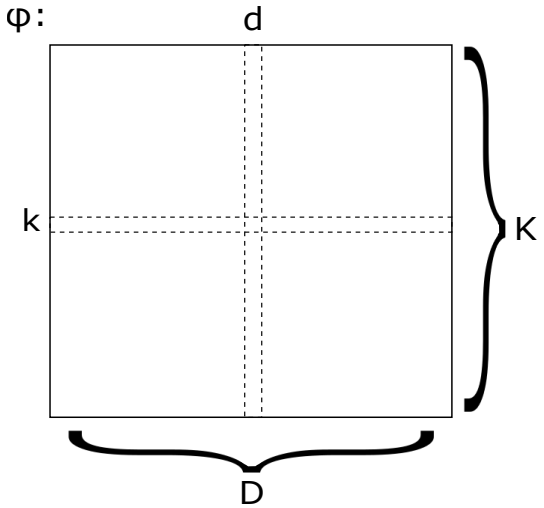
With our observables being: $X = \{x_1, \ldots, x_N\}$ the latent states being: $Z = \{z_1, \ldots, z_N\}$ and our model parameters: $\Theta = \{\pi, A, \phi\}$
The matrix A looks like this:



Where: $A_{jk} = p(z_{nk} = 1|z_{n-1,j} = 1) =$ probability of going from j to k
The matrix π looks like this:



Where: $\pi_k = p(z_{nk} = 1)$.
The matrix ϕ looks like this:



Where: $\phi_{kd} = p(x_{nd} = 1|z_{nk} = 1)$ = probability of seeing observation d in state k.

2.1.3 HMM as a generative model

There are two goals we might have when we are using HMMs, first is to determine the likelihood of a sequence of observations. The second is to find a plausible underlying explanation (or decoding) of a sequence of observations.

Determining the likelihood of a sequence of observations are computed like

this:

$$p(X|\Theta) = \sum_Z p(X, Z|\Theta)$$

This sum has K^N terms, but it turns out that it can actually be computed in $\mathcal{O}\left(K^2 N\right)$ time. But let's look at decoding first.

2.1.4 Decoding using HMMs

Given a HMM Θ and a sequence of observations $X = x_1, \ldots, x_N$, then find a plausible explanation $Z^* = z_1^*, \ldots, z_N^*$ of values of the hidden variable. We will look at two types of decoding here, Viterbi deconding and Posterior decoding.

Viterbi decoding Z^* is the overall most likely explanation of X:

$$Z^* = \arg\max_Z p(X, Z|\Theta)$$

Posterior decoding z_n^* is the most likely state to be in the n'th step:

$$z_n^* = \arg\max_{z_n} p(z_n|x_1, \ldots, x_N)$$

2.1.5 Viterbi decoding

Given X, find Z^* such that: $Z^* = \arg\max_Z p(X, Z|\Theta)$

We can write the probability of X given the optimal explanation Z^* as follows:

$$\begin{aligned}
p(X, Z^*) &= \max_Z p(X, Z) \\
&= \max_{z_1,\ldots,z_N} p(x_1, \ldots, x_N, z_1, \ldots, z_N) \\
&= \max_{z_N} \max_{z_1,\ldots,z_{N-1}} p(x_1, \ldots, x_N, z_1, \ldots, z_N) \\
&= \max_{z_N} \omega(z_N) \\
z_N^* &= \arg\max_{z_N} \omega(z_N)
\end{aligned}$$

Here, $\omega(z_n) = \max_{z_1,\ldots,z_{n-1}} p(x_1, \ldots, x_n, z_1, \ldots, z_n)$ is the probability of the most likely sequence of states $z_1, \ldots, z_n$ ending in z_n generating the observations $x_1, \ldots, x_n$.

We can expand $\omega(z_n)$ as follows:

$$\begin{aligned}
\omega(z_n) &= \max_{z_1,\ldots,z_{n-1}} p(x_1,\ldots,x_n,\, z_1,\ldots,z_n) \\
&= \max_{z_1,\ldots,z_{n-1}} p(z_1)\prod_{i=2}^{n} p(z_i|z_{i-1})\prod_{i=1}^{n} p(x_i|z_i) \\
&= p(x_n|z_n)\max_{z_1,\ldots,z_{n-1}} p(z_1)\prod_{i=2}^{n} p(z_i|z_{i-1})\prod_{i=1}^{n-1} p(x_i|z_i) \\
&\vdots \\
&= p(x_n|z_n)\max_{z_{n-1}} p(z_n|z_{n-1})\omega(z_{n-1})
\end{aligned}$$

So the ω is recursive, we can sum up ω as follows:

Recursion

$$\omega(z_n) = p(x_n|z_n)\max_{z_{n-1}} \omega(z_{n-1})p(z_n|z_{n-1})$$

Basis

$$\omega(z_1) = p(x_1, z_1) = p(z_1)p(x_1|z_1)$$

We can describe the ω function as a $N \times K$ matrix, $\omega[k][n] = \omega(z_n)$ if z_n is state k, and construct it as follows:

1. $\omega[k][n] = 0$
2. if $p(x[n]|k) \neq 0$:
   (a) for $j = 1$ to K
      i. if $p(k|j) \neq 0$
         A. $\omega[k][n] = \max(\omega[k][n], p(x[n]|k) \cdot \omega[j][n-1] \cdot p(k|j)$

We can then compute ω in $\mathcal{O}\left(K^2N\right)$ time, using $\mathcal{O}\left(KN\right)$ space. Now that we have $\omega(z_n)$, which is the probability of the most likely sequence of states, that ends in z_n, we can find Z^* by backtracking :

$$\begin{aligned}
z_N^* &= \arg\max_{z_N} \omega(z_N) = \arg\max_{z_N}\max_{z_{N-1}} \left(p(x_N|z_N)\omega(z_{n-1})p(z_N|z_{N-1})\right) \\
z_{N-1}^* &= \arg\max_{z_{N-1}} \left(p(x_N|z_N^*)\omega(z_{N-1}p(z_N^*|z_{N-1})\right) \\
z_{N-2}^* &= \arg\max_{z_{N-2}} \left(p(x_{N-1}|z_{N-1}^*)\omega(z_{N-2}p(z_{N-1}^*|z_{N-2})\right)
\end{aligned}$$

So we obviously have all that we need, we can just backtrack as follows:

1. $z[1\ldots N] = \text{undef}$
2. $z[N] = \arg\max_k \omega[k][N]$

3. $n = N - 1$ to 1:

   (a)
   $$z[n] = \arg\max_k \left(p(x[n+1]|z[n+1]) \cdot \omega[k][n] \cdot p(z[n+1]|k)\right)$$

4. print $z[1 \dots N]$

We can backtrack in time $\mathcal{O}(KN)$ using space $\mathcal{O}(KN)$ using ω.

We could also use "posterior decoding", but it only works if there is a transition from any state to any other state. In other cases it might not return syntactically correct results.

2.1.6 Computing the likelihood of a sequence of observations

Apart from the ω recursion, we have the forward and backward algorithms:

Forward algorithm Computes $\alpha(z_n)$ which is the joint probability of observing $x_1, \dots, x_n$ and being in state z_n:

$$\alpha(z_n) = p(x_1, \dots, x_n, z_n)$$

Backward algorithm Computes $\beta(z_n)$ which is the conditional probability of observing $x_{n+1}, \dots, x_N$ in the future, assuming we are in state z_n

$$\beta(z_n) = p(x_{n+1}, \dots, x_N | z_n)$$

Then, having $\alpha(z_n)$ and $\beta(z_n)$, we can get the likelihood of the observations as:

$$p(X) = \sum_{z_n} \alpha(z_n)\beta(z_n), \quad p(X) \sum_{z_N} \alpha(z_N)$$

We can compute α as:

Recursion
$$\alpha(z_n) = p(x_n|z_n) \sum_{z_{n-1}} \alpha(z_{n-1}) p(z_n|z_{n-1})$$

Basis
$$\alpha(z_1) = p(x_1, z_1) = p(z_1)p(x_1|z_1)$$

And we can compute β as:

Recursion
$$\beta(z_n) = \sum_{z_{n+1}} \beta(z_{n+1}) p(x_{n+1}|z_{n+1}) p(z_{n+1}|z_n)$$

Basis
$$\beta(z_N) = 1$$

2.1.7 Training HMMs