Notes on MAP Estimation for HMMs

Anoop Sarkar

A HMM is a probabilistic finite-state automata, in which $a_{q,p}$ represents the probability of taking a transition from state q to state p and $b_{q,k}$ represents the probability of emitting alphabet symbol k from state q. So our HMM is represented by $\theta = (a, b)$. Let us assume we have n states and m vocabulary items that can be emitted from any state. Both a_q and b_q for a given q are *multinomial* distributions: measuring the probability of choosing to transition to another state from state q or choosing to emit a vocabulary symbol when at state q.

Let's consider a_q in more detail, and everything we say in that case applies to b_q. The only difference is the transition outcome is from a set of states: $1, \ldots, n$, and the emission outcome is from the set of vocabulary items: $1, \ldots, m$.

We want to describe the probability of the data given a model. In this case, for a given state q the probability $a_{q,p}$ describes the probability of a possible transition from q to p. Let's say we observe c independent samples and we want to estimate the probability that the c samples are assigned by $a_{q,p}$. The probability assigned to c samples depends only on the counts of each outcome: i.e. each transition starting at q observed in the sample: $c_{1,q}, c_{2,q}, \ldots, c_{n,q}$ where $c = c_{1,q} + c_{2,q} + \ldots + c_{n,q}$. The probability of this observation is:

$$\Pr(c_{1,q}, c_{2,q}, \ldots, c_{n,q} \mid c, a_{q,p}) = \frac{c!}{c_{1,q}! \ldots c_{n,q}!} \prod_{p=1}^{n} (a_{q,p})^{c_{q,p}}$$

The factor $\frac{c!}{c_{1,q}! \ldots c_{n,q}!}$ is required because this is a distribution for *unordered samples*, where sequences of outcomes that are permutations on one another are considered to be the same joint event. For *ordered samples* the distribution is simply:

$$\Pr(c_{1,q}, c_{2,q}, \ldots, c_{n,q} \mid c, a_{q,p}) = \prod_{p=1}^{n} (a_{q,p})^{c_{q,p}}$$

The normal scenario in learning is that a fixed set of outcomes (a sample) is provided and what we care about estimating the probability $a_{q,p}$ in which case the difference between the ordered and unordered case is a constant so we can ignore it. We generally use the ordered samples case because it is simpler.

A Dirichlet prior is a prior distribution over each set of multinomial parameters in the HMM. The parameters at state q can be combined with this prior. Consider the prior probability of a_q:

$$g(a_q) = \frac{1}{B(\nu_{1,q}, \ldots, \nu_{n,q})} \prod_{p=1}^{n} (a_{q,p})^{\nu_{q,p}-1}$$

$B(\nu_{1,q}, \ldots, \nu_{n,q})$ is the n-dimensional Beta function,

$$B(\nu_{1,q}, \ldots, \nu_{n,q}) = \frac{\Gamma(\nu_{1,q}) \ldots \Gamma(\nu_{n,q})}{\Gamma(\nu_{1,q} + \ldots + \nu_{n,q})}$$

where $\Gamma(n) = (n-1)!$, assuming that $\nu_{1,q}, \ldots, \nu_{n,q}$ are all integers (which is the usual assumption). The prior expectation of a transition from q to p is $\frac{\nu_{q,p}}{\nu_0}$ where $\nu_0 = \sum_i \nu_{i,q}$.

Let $D = c_{1,q}, c_{2,q}, \ldots, c_{n,q}$ and $c = c_{1,q} + c_{2,q} + \ldots + c_{n,q}$. The posterior probability $P(a_q \mid D)$ is:

$$\begin{aligned}
P(a_q \mid D) &\approx P(D \mid a_q) \cdot P(a_q) \\
P(D \mid a_q) &= \prod_{p=1}^{n} (a_{q,p})^{c_{q,p}} \\
P(a_q) &= \frac{1}{B(\nu_{1,q}, \ldots, \nu_{n,q})} \prod_{p=1}^{n} (a_{q,p})^{\nu_{q,p}-1} \\
P(a_q \mid D) &= \prod_{p=1}^{n} (a_{q,p})^{c_{q,p}} \cdot \frac{1}{B(\nu_{1,q}, \ldots, \nu_{n,q})} \prod_{p=1}^{n} (a_{q,p})^{\nu_{q,p}-1} \\
&= \frac{1}{B(c_{1,q} + \nu_{1,q}, \ldots, c_{n,q} + \nu_{n,q})} \prod_{p=1}^{n} (a_{q,p})^{(\nu_{q,p}-1)+c_{q,p}}
\end{aligned}$$

Note that $P(a_q \mid D)$ is in the same form as the rhs of $P(a_q)$, and let's assume we want to re-estimate $P(D \mid a_q)$ iteratively, we can compute a new value for the posterior $P(a_q \mid D)$ by using this new estimate of $P(D \mid a_q)$ for the current iteration multiplied by the value of $P(a_q \mid D)$ from the last iteration as a new *conjugate prior* which provides a new value for $P(a_q)$ for the current iteration.

In practice, we set ν_i to be an integer greater than 1. If $\nu_i > 1$ and an integer then the prior simply reduces to adding $\nu_i - 1$ *virtual* samples to the likelihood expression, resulting in a MAP estimate for $a_{q,p}$ which is the simple expression (note how it looks just like smoothing!):

$$a_{q,p} = \frac{(\nu_{q,p} - 1) + c_{q,p}}{\sum_r (\nu_r - 1) + \sum_r c_{r,q}}$$

Note that for transition probabilities hyperparameters $\nu_{1,q}, \ldots, \nu_{n,q}$ can be tied to one value: ν_q^t, the hyperparameter for the transition probability from state q. Similarly, the emission hyperparameters can be all tied to a single value: ν_q^e. Alternatively each $\nu_{q,p}$ for transition and emission probabilities can be set individually based on prior knowledge.

The above explanation shows how MAP can be thought of as providing the basis for *smoothing* each probability estimated from the data. Notice that we already do this for the supervised training of our HMMs. All you need to do is to also add in the same smoothing terms when doing MAP estimation. In our case, the "virtual" counts and the estimates from the labeled data are used in each iteration of MAP and the new values for $\nu_{q,p}$ in each iteration is simply the value of $\nu_{q,p}$ from the previous iteration plus the (expected) counts $c_{q,p}$. So it turns out that doing MAP estimation correctly in the current implementation is simply a couple of additions away!

References

Peter Cheeseman, James Kelly, Matthew Self, John Stutz, Will Taylor, and Don Freedman. 1988. AutoClass: A Bayesian Classification System. In *Proc. of the 5th Int'l Conf. on Machine Learning* ICML-1988, pp. 54–64.

Gregory Cooper and Edward Herskovits. 1992. A Bayesian method for the induction of probabilistic networks from data. In *Machine Learning* 9:309–347.

Wray Buntine. 1992. Learning Classification Trees. In *Artificial Intelligence Frontiers in Statistics: AI and Statistics III*, ed. by D. J. "*Talk to the Hand!*" Hand. Chapman and Hall.

Andreas Stolcke. 1994. *Bayesian Learning of Probabilistic Language Models.* PhD Dissertation. University of California at Berkeley.