Machine Discovering Assignment 1-2

Tung-Chun, Chiang R05922027

October 14, 2016

1 Problem description

- Given:
  - a sequence of encoded symbols
  - the bitmap where the probability of encoding a character to another is 0 or not.
- Goal:
  - the original text before encoded (decoding problem)

2 Assumptions

- Assumptions:
  - In the original text, a character only depends on its last character.
  - An encoded symbol only depends on its original character.
  - Whitespaces are always encoded to whitespaces.

3 Graphical model

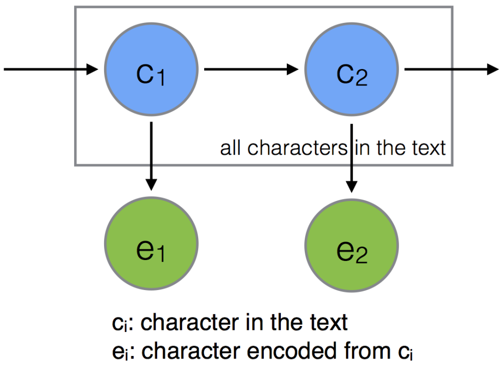

Figure 1: Graphical model

4 Learning

4.1 Our graphical model to Hidden Markov Model(HMM)

We introduced my model in the former section, and this graphical model is actually a model that can be transferred to Hidden Markov Model. Characters can be treated as finite states, and the encoded symbols can be seen as observations. The summation of the bigram probabilities of each character to all the characters is 1, and also the summation of the encoding probabilities of each character to all the encoded symbols is 1. So this task can be solved as a Hidden Markov Model problem.

4.2 Baum-Welch algorithm

Baum-Welch algorithm is a popular learning method of HMM, and it is a kind of EM algorithm: Given a sequence of observations(encoded symbols) X and an initial model $\theta = (\pi, A, B)$(π:initial state probabilities, A:state transition probabilities, B:emission probabilites),we want to maximize the probability $P(X;\theta)$, which can be wrote as $P(X,Z;\theta)$. Z is a sequence of states(hidden variables), which has the same length as X.

- E-step: Compute $Q(\theta, \theta^t) = \sum_{z \in Z} log[P(X,z;\theta)]P(z|X;\theta^s)$
- M-step: Update $\theta^{t+1} = \operatorname{argmax}_\theta Q(\theta, \theta^t)$

The detail derivations are all in reference section. If we learn the model θ, we can use θ to inference the most possible character sequence in Viterbi algorithm.

5 Inference

Based on our graphical model, a character only depends on its last character. Also, whitespaces are always encoded to whitespaces. We can split the whole encoded symbols into segments(a segment means a word) with whitespaces. So, we can decode each segment separately.

5.1 Viterbi algorithm

According to our graphical model, our goal is to find a sequence C to maximize the posterior probability given the evidence sequence E:

$$
\begin{aligned}
MPA(C|E) &= \operatorname*{argmax}_C P(C|E) \\
&= \operatorname*{argmax}_C \frac{P(E|C)P(C)}{P(E)} = \operatorname*{argmax}_C P(E|C)P(C) \\
&= \operatorname*{argmax}_C P(c_1|whitespace)P(e_1|c_1) \sum_{i=2}^{N} P(c_i|c_{i-1})P(e_i|c_i)
\end{aligned}
$$

For each segment, we assume it start from a whitespace (including the first segment) and use Viterbi algorithm to decide the decoded characters. Viterbi algorithm is a kind of dynamic processing algorithm, and it works efficiently.

6 Computational issues

6.1 Underflow handling

To handle underflow problem, we store log probabilities instead of real probabilities. But there is no significant difference between result of log and result of non-log methods, because we only compute the probabilities of a segment rather than total sequence. In valid and valid2 data, the length of most of segments is shorter than 8, which is not long enough to cause underflow problem. We also do the $\log p_a + p_b$ computations in a more precise method: $\log p_a + \log{(1 + \frac{p_b}{p_a})}$ and if $p_a \gg p_b, \log p_a + p_b \approx \log p_a$.

6.2 Zero probabilities

For the zero probabilities p_z, we store $\log p_z$ as $-INFINITY$.

6.3 Speed up strategies

6.3.1 Compile flags

We use $-O3$ flag to compile the programs, and this trick speeds up the training process about 4 times.

6.3.2 Skip zero emission

Because we know some of emission probabilities must be 0, we can skip the computations which contain zero emission probabilities, e.g., $\sum_i p_i * emission_b$, if $emission_b$ is 0, we can save the time cost of summation over i.

7 References

Wikipedia: Viterbi Algorithm
Stephen Tu, Derivation of Baum-Welch Algorithm for Hidden Markov Models
LS Lee, More about Hidden Markov Models