0: Summary

In the recitation we discussed the following topics:

(1) What a Markov Model is.
(2) What a Hidden Markov Model is.
(3) Presented what are the main three problems concerning a HMM.
(4) We presented solutions for the two first, i.e., every problem except for the problem of training a HMM.

You will learn more about it when we see the EM algorithm (scheme of algorithms actually) in class.

I strongly suggest everyone reading Rabiner’s tutorial since that was the main source that the recitation was based on. Actually this set of notes, is a companion -written a bit less informal compared to previous sets of notes- to help you study the part of Rabiner’s tutorial that we discussed in class. I use the following colors to encode the following different types of remarks:

(1) Magenta: **FOOD FOR THOUGHT:** in order to provoke your thought. For example in the midterm, I am sure many of you could do much better if you had sufficient time. You can save time in the final by asking yourself questions throughout the semester. Make sure you know how to answer the questions your are given here.
(2) Blue: What you should read from Rabiner’s tutorial.
(3) Red: General Remarks.

1. Markov Model

**Definition 1.** A Markov model is defined by the following components:

1. A set of states \( S = \{ s_1, \ldots, s_n \} \).
2. A starting distribution \( \pi = [\pi_1, \ldots, \pi_n] \), where \( \pi_j = \Pr(q(0) = s_j) \), i.e., the probability of starting from the j-th state.
3. A \( n \times n \) probability transition matrix \( P = [p_{ij}] \) where \( p_{ij} = \Pr(q(n + 1) = s_j | q(n) = s_i) = p_{ij} \) for every \( n=0,1,.. \).

It is easy to simulate a Markov Model:

1. You start from a given state according to the starting probability distribution \( \pi \).
2. You pick the next state using the transition matrix \( P \), given the prior state.

*Date: 12 November 2009.*
(3) Repeat the above two steps.

A Markov Chain is a sequence of states generated according to the Markov Model. For example see figure [1] The figure visualizes the transition matrix $P$. With a starting probability distribution, we have a simple Markov Model modeling weather transitions.

If $\pi = [0.6, 0.4]$ then a simple simulation in matlab would be the following:

MATLAB CODE

```matlab
MAXITER=100; %<-- let’s simulate it for few iterations
q = zeros(MAXITER+1,1);

q(1)='s';

for i = 2 : MAXITER+1
    tmp = rand; %<-- draw from uniform(0,1)
    if (tmp <= .6) %<-- pi = [0.6 0.4]
        q(i)='s';
    else
        q(i)='r';
    end
end

fprintf('Simulated Markov Chain: %s 
',q);
```

There is a huge literature behind Markov Models with numerous application. Actually Markov Models play a prominent role in Bayesian inference where a common task is to sample from “ugly” probability distributions.

After Sue’s question we also mentioned that Markov Models play a prominent algorithm in Google’s ranking algorithm (at least the published algorithm of Pagerank).
We will not say more but for those interested to learn more on Markov Models you can check the following links (Completely Optional):


If many of you are interested on this, i.e., sampling using Markov Models, please mail me and we can schedule an extra recitation. It is totally related to the class, even if it is not in the syllabus.

Food for thought: Assume that one gives you data generated by the above simulated Markov Chain but you do not know that. How would you decide a model for the data, i.e., if the weather transitions are independent or follow a HMM model? If you have hard time answering this, read more on Model Selection!

2. Hidden Markov Model

So, the main focus of the recitation was the HMM and problems related to it. Before going into the formal definition, let’s elaborate on words: just by the words Hidden Markov Model we understand that it should be a Markov Model (and this is exactly why we had the short discussion on Markov Models) but it is “hidden”. But what does it refer to? Actually, the word hidden refers to the fact that we don’t get to observe the states of the Markov Model, but a set of symbols. However, what we get to observe is influenced by the Markov Model. It time to see the formal definition (written a bit informally though)

Definition 2. A Hidden Markov Model (HMM) is a Markov Model thus it is comprised by:

- A set of states $S = \{s_1, \ldots, s_n\}$.
- A starting distribution $\pi = [\pi_1, \ldots, \pi_n]$, where $\pi_j = \Pr(q(0) = s_j)$, i.e., the probability of starting from the j-th state.
- A $n \times n$ probability transition matrix $P = [p_{ij}]$ where $p_{ij} = \Pr(q(n+1) = s_j | q(n) = s_i) = p_{ij}$ for every $n=0,1,..$

But it is also hidden! We thus have the following elements that complete the definition of a HMM:

- A set of possible outputs $\Sigma = \{\sigma_1, \ldots, \sigma_m\}$. We call $\Sigma$ also alphabet in the following.
- A $n \times m$ emission probability matrix, i.e., output probability matrix, $B$, where $b_{ij} =$ Probability of emitting symbol $\sigma_j$ in state $i$. Rabiner uses the notation $b_i(j)$ for the element $b_{ij}$ of matrix B.
Back to our example of figure 1. Let’s define that our alphabet consists of three symbols: \( \Sigma = \{ \alpha, \beta, \gamma \} \). Therefore matrix \( B \) should be a \( 2 \times 3 \) matrix. What you would observe now would be a sequence of Greek letters rather than sequence of s’s and r’s. We said that the simulation is easy and needs only the rand() function of MATLAB, the pseudocode can be found in Rabiner’s tutorial, page 261. Try to simulate a HMM of your own choice in analogy with the above MATLAB code, for your own purposes (and then apply some of the algorithms that you implemented, i.e., make sure things make sense in synthetic data).

As we said there are three main problems concerning a HMM. These are described in Rabiner’s tutorial, page 261.

2.1. **Problem 1: Given your HMM and a sequence of observations \( O = O_1 \ldots O_T \), compute \( \Pr(O) \).** This sounds as a very easy problem since we can just sum over all possible sequences of states the probability of the observed sequence \( O \). As we said this is exponential in \( T \), the number of observations which is really bad. However, we can use the definition of the HMM to come up with an efficient algorithm:

★ SOLUTION:. Rabiner’s tutorial, page 262-263. As outlined in Rabiner’s tutorial, the \( \beta \)s are not actually used for this computation, just the \( \alpha \)s.

2.2. **Problem 2: Given your HMM and a sequence of observations \( O = O_1 \ldots O_T \), find the most likely sequence of states.**

★ SOLUTION:. Forward-Backwards algorithm: Rabiner’s tutorial, page 263-264.

★ SOLUTION:. Viterbi’s algorithm: Rabiner’s tutorial, page 264.

**Food for thought:** Derive formulae (20), (25), (33a), if you have hard time after trying, mail the author of this document or come to office hours.

**Comment:** As we outlined in the recitation, the solutions are simple dynamic programming algorithms. The main key in dynamic programming algorithms is to find the correct subproblems, i.e., which lead us in the case of HMMs to define the key quantities \( \alpha_i(t), \beta_i(t), \gamma(t), \delta_i(t) \). For those of you not familiar with Dynamic Programming, you should work out first some simple examples and see why we defined the above quantities. For more on DP, you can check Chapter 15 [http://tinyurl.com/dpclrs](http://tinyurl.com/dpclrs).