Department of Mathematics and Computer Science
University of Southern Denmark, Odense

March 11, 2013
Marco Chiarandini

DM825 - Introduction to Machine Learning

Sheet 13, Spring 2013

Exercise 1 – Building Models
Construct a graphical model representing the following situation:

> Milk from a cow may be infected. To detect whether the milk is infected, you have a test, which may give either a positive or a negative test result. The test is not perfect. It may give a positive result on clean milk as well as a negative result on infected milk.
>
> From one day to another, the state of the milk can change. Cows with infected milk will heal over time, and a clean cow has a risk of having infected milk the next day. Now, imagine that the farmer performs the test each day. After a week, he has not only the current test result but also the six previous test results.
>
> It is good to have as a rule that no test is perfect. Unless you explicitly know otherwise, a test should always be given a positive probability of false positives as well as false negatives.

Exercise 2 – Building Models
Represent with a graphical model the possible types of failures that may incurr in this example:

> Consider an HIV test with a probability of false positives of 10^{-5}, and assume that a person has received a positive test result. Now, you may have the option of repeating the test, but will this be of any help?

Exercise 3 – Sequential data
Every day that he leaves work, Albert the Absent-minded Professor, toggles his light switch according to the following protocol: (i) if the light is on, he switches it off with probability 0.80; and (ii) if the light is off, he switches it on with probability 0.30. At no other time (other than the end of each day) is the light switch touched.

1. Suppose that on Monday night, Albert's office is equally likely to be light or dark. What is the probability that his office will be lit all the other four nights of the week (Tuesday through Friday)?
2. Suppose that you observe that his office is lit on both Monday and Friday nights. Compute the expected number of nights, from that Monday through Friday, that his office is lit.

Now suppose that Albert has been working for five years (i.e., assume that the Markov chain is in steady state).

3. Is his light more likely to be on or off at the end of a given workday?

Exercise 4 – Bayesian Networks – Practical

There are many software for dealing with probabilistic graphical models. They differ in many characteristics, among the others, tasks and formal language. The following is an incomplete list:

- Hugin: a commercial system made in Denmark
- OpenBugs: Bayesian inference Using Gibbs Sampling. The package BRugs allows OpenBUGS analyses to be run fully interactively from within R.
- Mallet (MAchine Learning for LanguagE Toolkit) is a Java-based package for statistical natural language processing in *document classification*.
- Bayes Net package in Weka: implements various Bayesian network classifier learning algorithms.
- In R under the CRAN Task View, gR Graphical Models Task lists all packages for graphical models. Among them `deal`.

In the following it is reported an example in `deal`. Carry out, if possible, the same analysis in BRugs.

```
data(ksl)
?ksl
head(ksl)
str(ksl)
summary(ksl)

# define the nodes and type of variables corresponding to them
# in the Bayesian network
(nw <- network(ksl)) # network does not use the data

## Attach prior independent probability distributions
localprob(nw,"FEV") <- c(179,80)
localprob(nw,"Kol") <- c(691,80)
localprob(nw,"Hyp") <- c(0.55,80)
localprob(nw,"logBMI") <- c(3.23,80)
localprob(nw,"Smok") <- c(0.5,0.5)
localprob(nw,"Alc") <- c(0.5,0.5)
localprob(nw,"Work") <- c(0.5,0.5)
localprob(nw,"Sex") <- c(0.5,0.5)
localprob(nw,"Year") <- c(0.5,0.5)

plot(nw) # plots the network. No arc are yet defined

# learn the probabilities from data
fit.prior <- jointprior(nw)
fit.learn <- learn(nw, ksl, fit.prior)
fit.nw <- getnetwork(fit.learn)
```

```
print(fit.nw, condposterior=TRUE)
plot(fit.nw)

# learn the conditional independency structure
fit.arcs <- getnetwork(autosearch(fit.nw,ksl,fit.prior))

print(fit.arcs, condposterior=TRUE)

# Alternatively, draw the arcs yourself
draw.nw <- getnetwork(drawnetwork(nw,ksl,fit.prior))
plot(draw.nw)

# PriorSampling or ancestral sampling from the network
rnetwork(makesimprob(fit.arcs),n=10)
```

Exercise 5 – Hidden Markov Models – Practical.
Consider the following example from [B3]. Your teacher wants to know whether or not you are actually preparing for the exam. There are four things you can be do on evenings {pub, TV, party, study} and he wants to work out whether or not you are studying. However, he cannot just ask you but he can make observations about your behavior and appearance. Specifically, he can observe if you look {Tired, Hungover, Scared, Fine}. He cannot know why you look the way you look but he has a probabilistic model to infer what is the probability that you look the way you look given what you did the last night. These probabilities are collected in an *emission probability table*.
The other information he uses are the probabilities that you are in a state tonight given that you were in another state the last night. These probabilities are collected in a *transition probability table*.
On the basis of his experience as a student he makes the following guesses on the probability tables: for the transition probabilities

	Previous night			
	TV	Pub	Party	Study
TV	0.4	0.6	0.7	0.3
Pub	0.3	0.05	0.05	0.4
Party	0.1	0.1	0.05	0.25
Study	0.2	0.25	0.2	0.05

and for the observation probabilities:

	TV	Pub	Party	Study
Tired	0.4	0.6	0.7	0.3
Hungover	0.3	0.05	0.05	0.4
Scared	0.1	0.1	0.05	0.25
Fine	0.2	0.25	0.2	0.05

After one week your teacher has collected the following observations (Tired, Tired, Hungover, Hungover, Scared, Hungover, Fine) and there are three things that he might want to do with these data:

- see how well the sequence of observations that he has made match his current probability tables, ie, what is the probability of that joint outcome of observations given his model.

- work out the most probable sequence of states that you have been in based on his observations.
- given several sets of observations (for example, by watching several students) generate a good model from the data.

Write these questions in the formal way of probability calculus. Describe the algorithmic issues for answering them. Carry out the computations in R using one of the two packages above.
Below are the data ready to be imported in R. They include the probability of the initial and final states, the tables above and the observations.

```
aFirst <- c(0.25,0.25,0.25,0.25)
aLast <- c(0.25,0.25,0.25,0.25)
a <- matrix(c(.4,.3,.1,.2,.6,.05,.1,.25,.7,.05,.05,.2,.3,.4,.25,.05),ncol
    =4,byrow=FALSE)
b <- matrix(c(.2,.1,.2,.5,.4,.2,.1,.3,.3,.4,.2,.1,.3,.05,.3,.35),ncol=4,
    byrow=FALSE)
obs <- c(0,0,3,1,1,2,1,3)
```

```
HMMfwd(a,b,obs)
Viterbi(a,b,obs)
ViterbiSimple(a,b,obs)
BaumWelch(obs,4)
ViterbiSimple(a,b,obs)
```

There are several packages in R that implement Hidden Markov Models. Among them `HMM` is a simple implementation for discrete state and discrete space (ie, emission distribution). The package `HiddenMarkov` is a more general implementation that allows to declare the emission distribution for continuous and discrete variables.
Install and explore the possibilities of the two packages.

```
library(HMM)
?dishonestCasino
dishonestCasino()
```

```
library(HybridMarkov)
?HybridMarkov
?dthmm
demo("norm",package="HiddenMarkov")
```

- *Forward* algorithms are used for computing the conditional probability for a sequence of observations. It can tell how well a sequence of observations match with the current HMM.
- *Viterbi* algorithm work out the most probable sequence of states (latent variables) that match the observations
- *Baum-Welch or Forward-Backward* algorithm estimates (learns) the parameters of the transition and emission probabilities from the data given several sets of observations. It consists in an implementation of the EM algorithm together with the sum-product algorithm studied for polytrees.