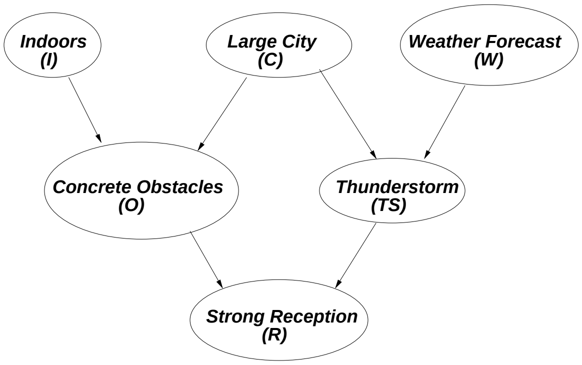
CSE 6369 - *Reasoning with Uncertainty*

Project 1- Fall 2015

Due Date: Oct. 20 2015

Bayesian Networks

1. Below is a probabilistic network to assess the probability of good radio reception. The goal is to use the available information about the receiver's location and the weather forecast to predict the probability of a good reception. According to this model, the probability of a good reception (R) depends on the presence of obstacles (O) and thunderstorms (TS). These, in turn, depend on the location (C and I) and on the weather forecast (W).



$$\begin{aligned}
R:\quad & P(R \mid O \wedge TS) = 0.2\\
& P(R \mid O \wedge \neg TS) = 0.4\\
& P(R \mid \neg O \wedge TS) = 0.5\\
& P(R \mid \neg O \wedge \neg TS) = 0.8
\end{aligned}$$

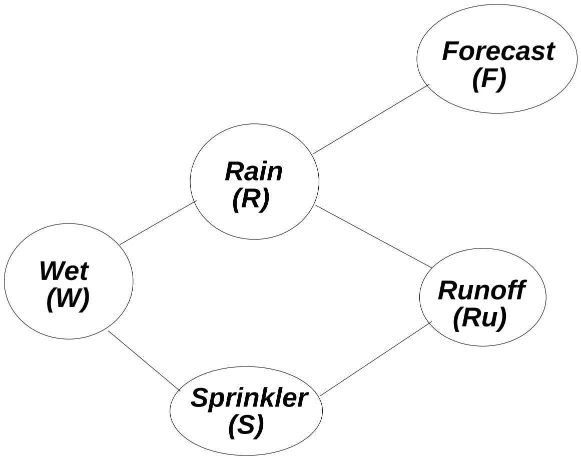
$$\begin{aligned}
O:\quad & P(O \mid I \wedge C) = 0.7\\
& P(O \mid I \wedge \neg C) = 0.4\\
& P(O \mid \neg I \wedge C) = 0.5\\
& P(O \mid \neg I \wedge \neg C) = 0.1\\
TS:\quad & P(TS \mid C \wedge W) = 0.6\\
& P(TS \mid C \wedge \neg W) = 0.05\\
& P(TS \mid \neg C \wedge W) = 0.5\\
& P(TS \mid \neg C \wedge \neg W) = 0.1\\
I:\quad & P(I) = 0.6\\
C:\quad & P(C) = 0.7\\
W:\quad & P(W) = 0.2
\end{aligned}$$

a) Manually compute the probability of there being a thunderstorm (TS) for the scenario $\neg O \wedge R$.

b) Implement likelihood weighted sampling for this network and the scenario listed in problem $a)$. Use this to re-compute the solution to problem $a)$ and also to compute the likelihood of being indoors in this same scenario.

Markov Networks

1. Consider the following Markov Network and corresponding clique potentials:



$Clique\ R, F:$

R	F	$\phi_{R,F}$
T	T	4.0
T	F	1.0
F	T	2.0
F	F	3.0

$Clique\ W, R:$

W	R	$\phi_{W,R}$
T	T	4.0
T	F	3.0
F	T	0.5
F	F	5.0

$Clique\ Ru, R:$

Ru	R	$\phi_{Ru,R}$
T	T	3.0
T	F	2.0
F	T	1.0
F	F	4.0

	S	W	$\phi_{S,W}$
$Clique\ S,W$:	T	T	4.0
	T	F	0.5
	F	T	2.0
	F	F	5.0

	S	Ru	$\phi_{S,Ru}$
$Clique\ S,Ru$:	T	T	3.0
	T	F	3.0
	F	T	2.0
	F	F	5.0

a) Implement Gibbs sampling for inference on this network.

b) For the scenario that there is rain in the forecast and the sprinklers are not running ($F \wedge \neg S$) use it to compute the likelihood that there is Runoff.

Kalman Filter

For this Project you have to implement a 2D Kalman filter to estimate the position of a square air-hockey puck moving in a rectangular box from noisy observations of the puck's position (the variance of the observation error in each direction is 0.01). To do this you are provided with a C library containing a simulator. You can implement your code directly in the file *Kalman_filter.c* which contains the function *kalman_filter(obs_pos)* that is directly called by the simulator and is provided with the noisy observation of the position. Alternatively you can use the simulator to generate a data file containing the observations (and the true location for comparison using the included act_pos and act_vel functions) and use this file in your program. The simulator code can be downloaded for Linux, Cygwin on Windows, and OSX from the course web page.
To start with this assignment you have to download the code, uncompress and untar it (*tar -xzvf PROJ1_machtype.tgz*). The resulting directory contains the following files:

Imakefile This file is used to create a machine specific Makefile by typing *xmkmf*.

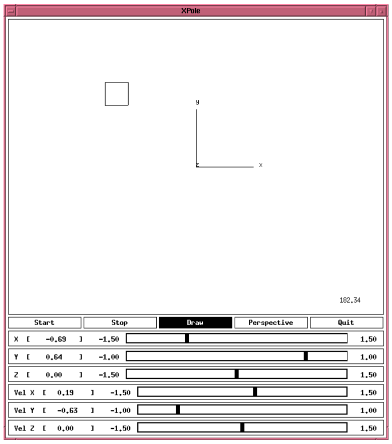
Kalman_filter.c This is the file you have to edit in order to implement the Kalman filter.

Kalman_filter.h Header file for the Kalman filter.

lib/libCube.a This library contains the simulator.

The Simulator

To generate the simulator you just have to type *make*. This creates the simulator executable *XCube*. The simulator should look as follows:



```
X     [  -0.69  ]  -1.50   1.50
Y     [   0.64  ]  -1.00   1.00
Z     [   0.00  ]  -1.50   1.50
Vel X [   0.19  ]  -1.50   1.50
Vel Y [  -0.63  ]  -1.00   1.00
Vel Z [   0.00  ]  -1.50   1.50
```

The *Perspective* button at the top of the simulator window will open a set of sliders which allow you to change your viewing angle (the default perspective is a top view).

3. Construct and implement a Kalman filter for the 2-dimensional tracking problem. For the system model you should assume that the object is moving at constant velocity.

   Record the difference between actual position and your filter estimate over time and plot its behavior for 3 different uncertainty assumptions for the system model. Also, analyze how the variances of the filter change over time.

   For this project you are to hand in your Kalman filter code (make sure your code is documented) and a short discussion of your experience with different uncertainty assumptions (include your Kalman filter equations and plots showing the change in error between actual position and the estimate and of the filter variance).

NOTE: You can find the formulas for the simple matrix inversion of a 2x2 matrix at
http://www.mathcentre.ac.uk/students.php/all_subjects/matrices/matrix_arithmetic/resources/56