

5.2:

- a. This problem can be solved using the filtering formula 15.5:

$$P(X_{t+1}|e_{1:t+1}) = \alpha P(e_{t+1}|X_{t+1}) \sum_{x_t} P(X_{t+1}|x_t) P(x_t|e_{1:t})$$

And replacing X with R and e with U to get:

$$P(R_{t+1}|u_{1:t+1}) = \alpha P(u_{t+1}|R_{t+1}) \sum_{r_t} P(R_{t+1}|r_t) P(r_t|u_{1:t})$$

When the convergence is achieved, we have:

$$abs(P(R_{t+1}|u_{1:t+1}) - P(R_t|u_{1:t})) < \varepsilon$$

Which becomes

$$P(R_{t+1}|u_{1:t+1}) = P(R_t|u_{1:t}) + \varepsilon \text{ where } \varepsilon > 0 \text{ is significantly small}$$

This can be approximated as

$$P(R_{t+1}|u_{1:t+1}) = P(R_t|u_{1:t})$$

Using the transition model to represent

$$(R_{t+1}|r_t) \text{ as } < 0.7, 0.3 > \text{ or } < 0.3, 0.7 >$$

and using the sensor model to represent

$$P(u_{t+1}|R_{t+1}) \text{ as } < 0.9, 0.2 > ,$$

we have:

$$P(R_{t+1}|u_{1:t+1}) = \alpha < 0.9, 0.2 > (< 0.7, 0.3 > P(r_t|u_{1:t}) + < 0.3, 0.7 > 1 - P(r_t|u_{1:t}))$$

Since

$$P(R_{t+1}|u_{1:t+1}) = P(R_t|u_{1:t})$$

then

$$P(R_{t+1}|u_{1:t+1}) = \alpha < 0.9, 0.2 > (< 0.7, 0.3 > P(r_{t+1}|u_{1:t+1}) + < 0.3, 0.7 > 1 - P(r_{t+1}|u_{1:t+1}))$$

$$P(R_{t+1}|u_{1:t+1}) = \alpha < 0.9, 0.2 > (< 0.4 * P(r_{t+1}|u_{1:t+1}), -0.4 * P(r_t|u_{1:t+1}) > + < 0.3, 0.7 >)$$

separating component wise and using $P(r_{t+1}|u_{1:t+1}) = p_r$:

$$p_r = \alpha(0.36p_r + 0.27)$$

$$(1 - p_r) = \alpha(-0.08p_r + 0.14)$$

thus we get:

$$\alpha(0.36p_r + 0.27) = 1 - \alpha(-0.08p_r + 0.14)$$

$$\alpha 0.36p_r + \alpha 0.08p_r = 1 - \alpha 0.14 - \alpha 0.27$$

$$\alpha 0.44p_r = 1 - \alpha 0.41$$

$$p_r = (1 - \alpha 0.41)/(\alpha 0.44)$$

and we have, by Bayes Rule:

$$\alpha = 1/((P(u_{t+1}|r_{t+1}))(P(r_{t+1}|r_t) p_r + ((1 - P(r_{t+1}|r_t)) * (1 - p_r))))$$

$$+ ((1 - P(u_{t+1}|r_{t+1}))((1 - P(r_{t+1}|r_t)) p_r + (P(r_{t+1}|r_t) * (1 - p_r))))$$

$$\alpha = 1/((0.9 * (0.7p_r + 0.3(1 - p_r))) + 0.2(0.3p_r + (0.7 * (1 - p_r))))$$

$$\alpha = 1/(0.9 * (0.4p_r + 0.3) + 0.2(-0.4p_r + 0.7))$$

$$\alpha = 1/(0.36p_r + 0.27 - 0.08p_r + 0.14)$$

$$\alpha = 1/(0.28p_r + 0.41)$$

$$p_r = \alpha(0.36p_r + 0.27)$$

$$p_r = \alpha(0.36p_r + 0.27)$$

$$(0.28p_r + 0.41)p_r = 0.36p_r + 0.27$$

$$0.28p_r^2 + 0.41p_r - 0.36p_r - 0.27 = 0$$

$$0.28p_r^2 + 0.05p_r - 0.27 = 0$$

using the quadratic formula

$$(-0.05 \pm \sqrt{0.05^2 - (4 * 0.28 * -0.27)}) / 2(0.28)$$

and using only the addition result in order to have positive probability, we have:

$$P(r_{t+1}|u_{1:t+1}) = p_r \approx 0.89674554944846594496715149092767$$

$$1 - P(r_{t+1}|u_{1:t+1}) = 1 - p_r \approx 0.10325445055153405503284850907233$$

b. Using the equation 15.6, we have

$$P(R_{2+k}|U_1, U_2) = P(R_{2+k}|r_{1+k})P(r_{1+k}|U_1, U_2) + (1 - P(R_{2+k}|r_{1+k}))(1 - P(r_{1+k}|U_1, U_2))$$

$$P(R_{2+k}|U_1, U_2) = <0.7, 0.3> P(r_{1+k}|U_1, U_2) + <0.3, 0.7> (1 - P(r_{1+k}|U_1, U_2))$$

$$P(R_{2+k}|U_1, U_2) = <0.7, 0.3> P(r_{1+k}|U_1, U_2) + <0.3, 0.7> (1 - P(r_{1+k}|U_1, U_2))$$

$$P(r_{2+k}|U_1, U_2) = 0.4 * P(r_{1+k}|U_1, U_2) + 0.3$$

Since the probability converges, we have

$$P(r_{2+k}|U_1, U_2) = P(r_{1+k}|U_1, U_2)$$

Thus:

$$P(r_{2+k}|U_1, U_2) = 0.4 * P(r_{2+k}|U_1, U_2) + 0.3$$

$$0.6 * P(r_{2+k}|U_1, U_2) = 0.3$$

$$P(r_{2+k}|U_1, U_2) = 0.5 = P(r_{1+k}|U_1, U_2)$$

Plugging this back into the equation:

$$P(r_{2+k}|U_1, U_2) = 0.4 * 0.5 + 0.3$$

$$P(r_{2+k}|U_1, U_2) = 0.2 + 0.3$$

$$P(r_{2+k}|U_1, U_2) = 0.5$$

So

$$P(r_{2+k}|U_1, U_2) \text{ converges to } 0.5$$

5.13 The variables are:

S to represent whether the students gets enough sleep or not

RE to represent whether the student has red eyes or not

SC to represent whether the student sleeps in class or not

As a DBN, we have

$$P(S_0) = 0.7$$

$$P(S_t|s_{t-1}) = <0.8, 0.3>$$

$$P(RE_t|x_t) = < 0.2, 0.7 >$$

$$P(SC_t|x_t) = < 0.1, 0.3 >$$

For an HMM, we have: ($\mathbf{1} = [1 \ 0; 0 \ 1]$)

$$\mathbf{T} = P(S_t|S_{t-1}) = [0.8 \ 0.3; 0.3 \ 0.8]$$

$$ORE_{true} = P(re_t|S_t) = [0.2 \ 0.0; 0.0 \ 0.7], \quad ORE_{false} = 1 - P(re_t|S_t) = [0.8 \ 0.0; 0.0 \ 0.3]$$

$$OSC_{true} = P(sc_t|S_t) = [0.1 \ 0.0; 0.0 \ 0.3], \quad OSC_{false} = 1 - P(sc_t|S_t) = [0.9 \ 0.0; 0.0 \ 0.7]$$

combining the observations into a single node, we have

$$OBS_{true} = P(re_t, sc_t|S_t) = [0.02 \ 0.0; 0.0 \ 0.21]$$

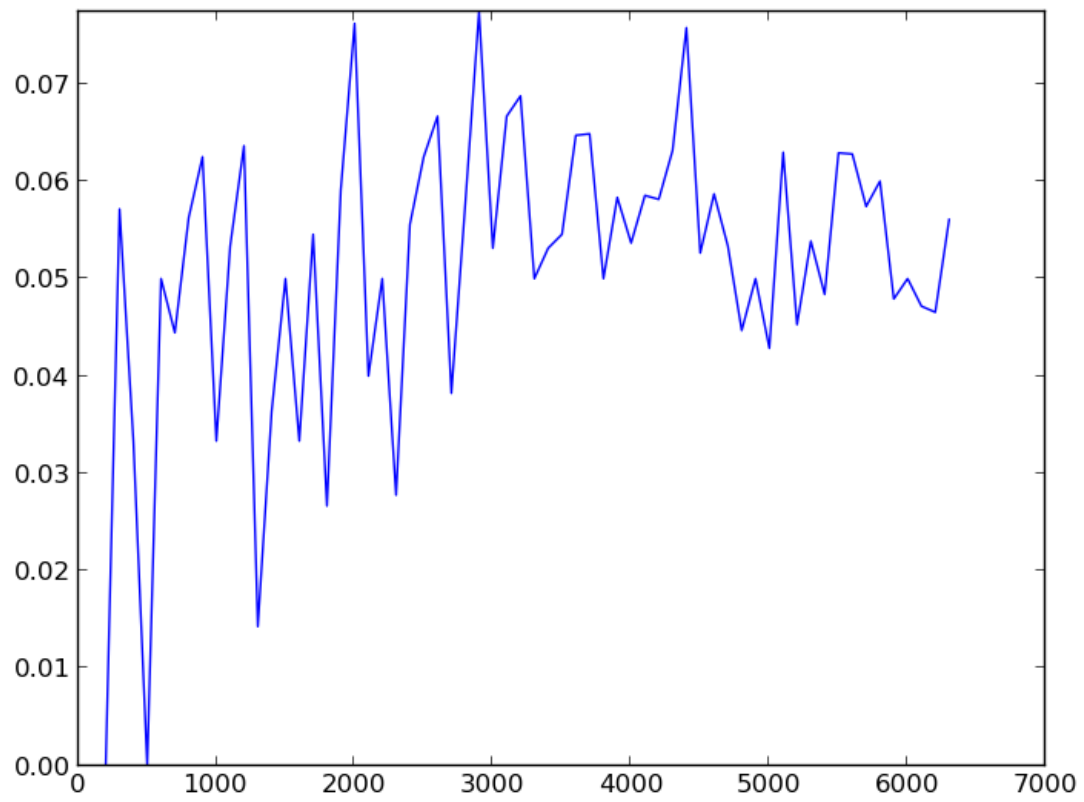
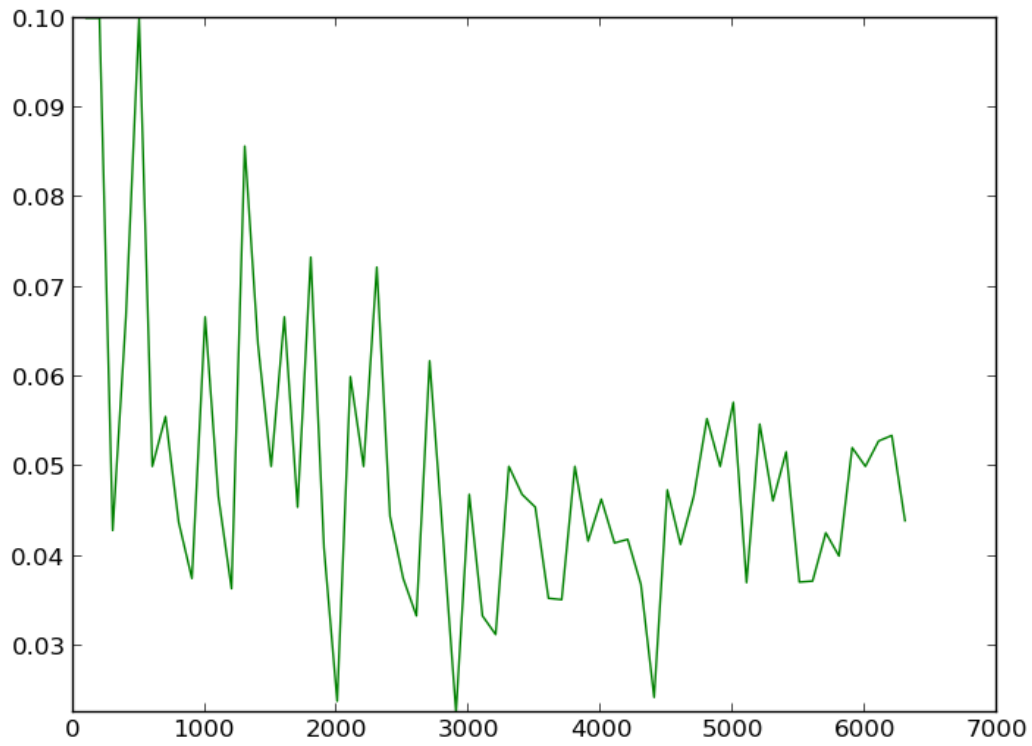
$$OBS_{false} = 1 - P(re_t, sc_t|S_t) = [0.98 \ 0.0; 0.0 \ 0.79]$$

Sampling Writeup:

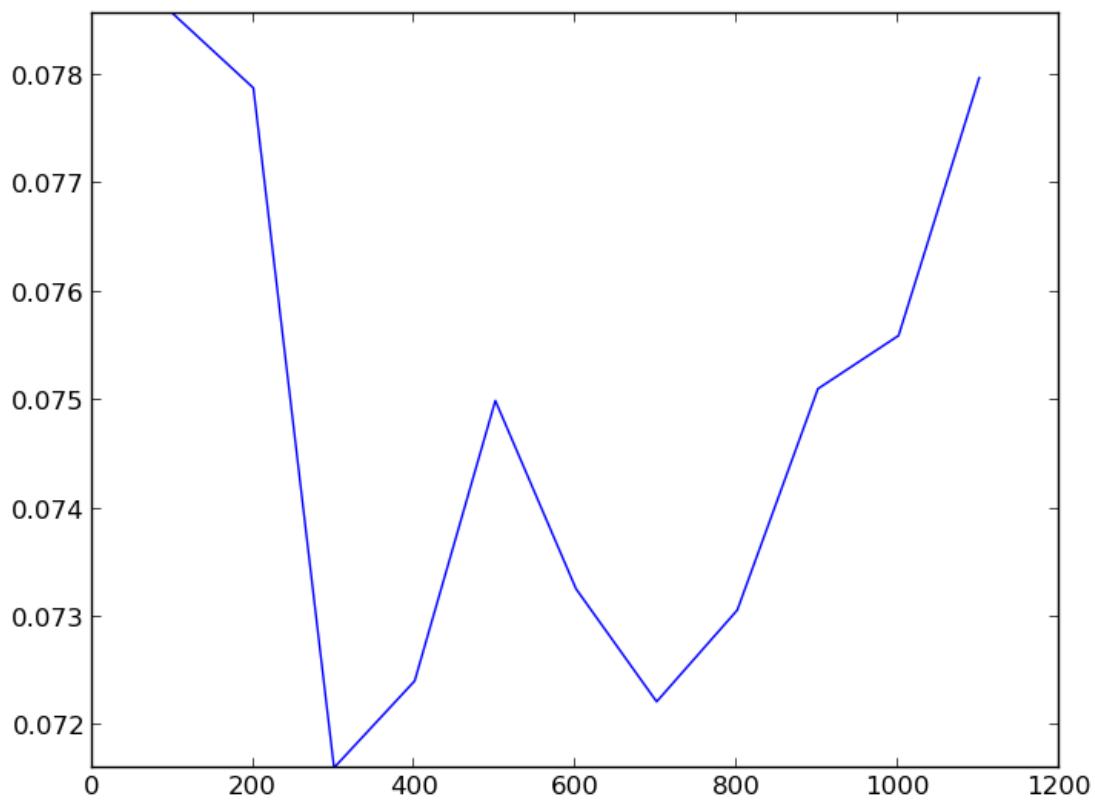
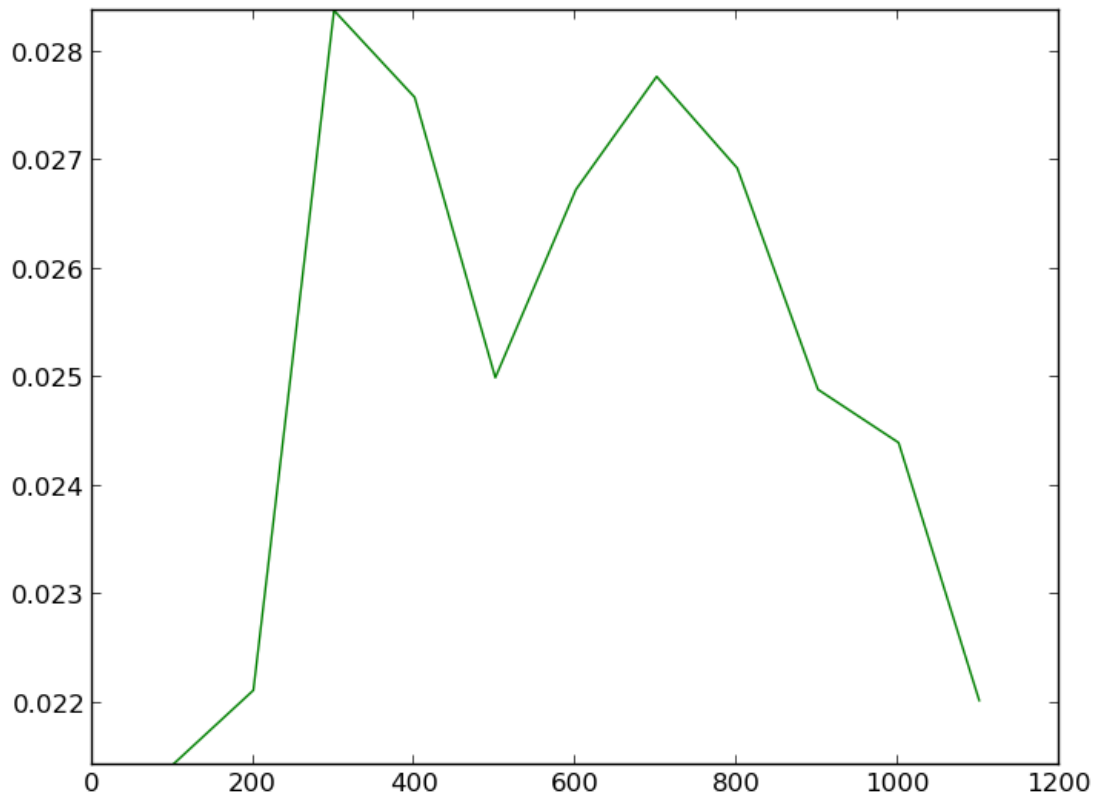
1. Rejection Sampling Graphs:

Green lines represent True values, Blue lines for False values,
X axis represents the number of samples, Y axis for the average values

q, -, t, f Graphs:

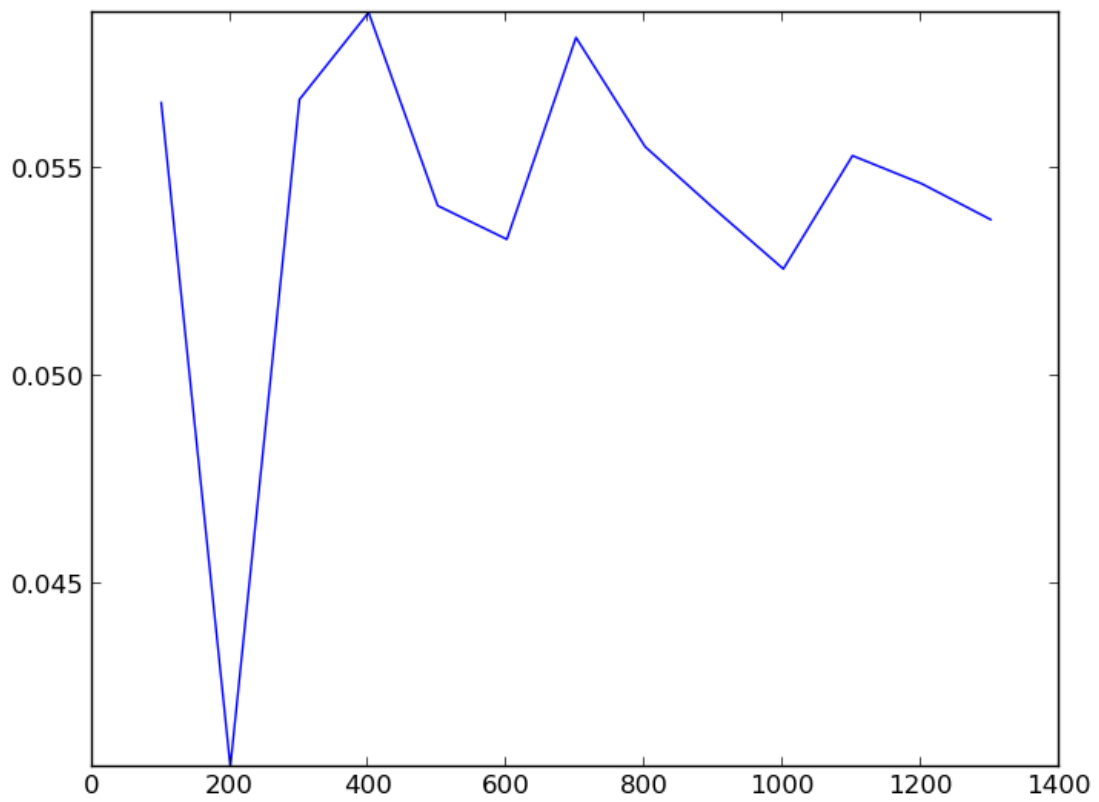
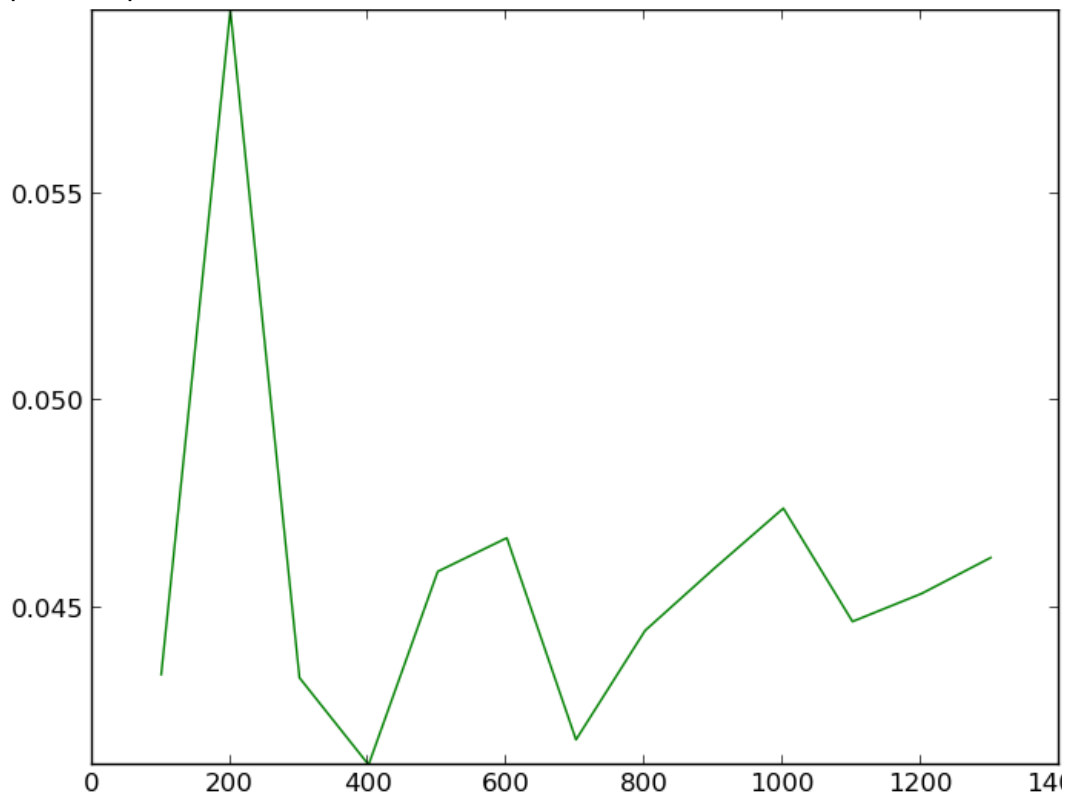


t,-,q Graphs:

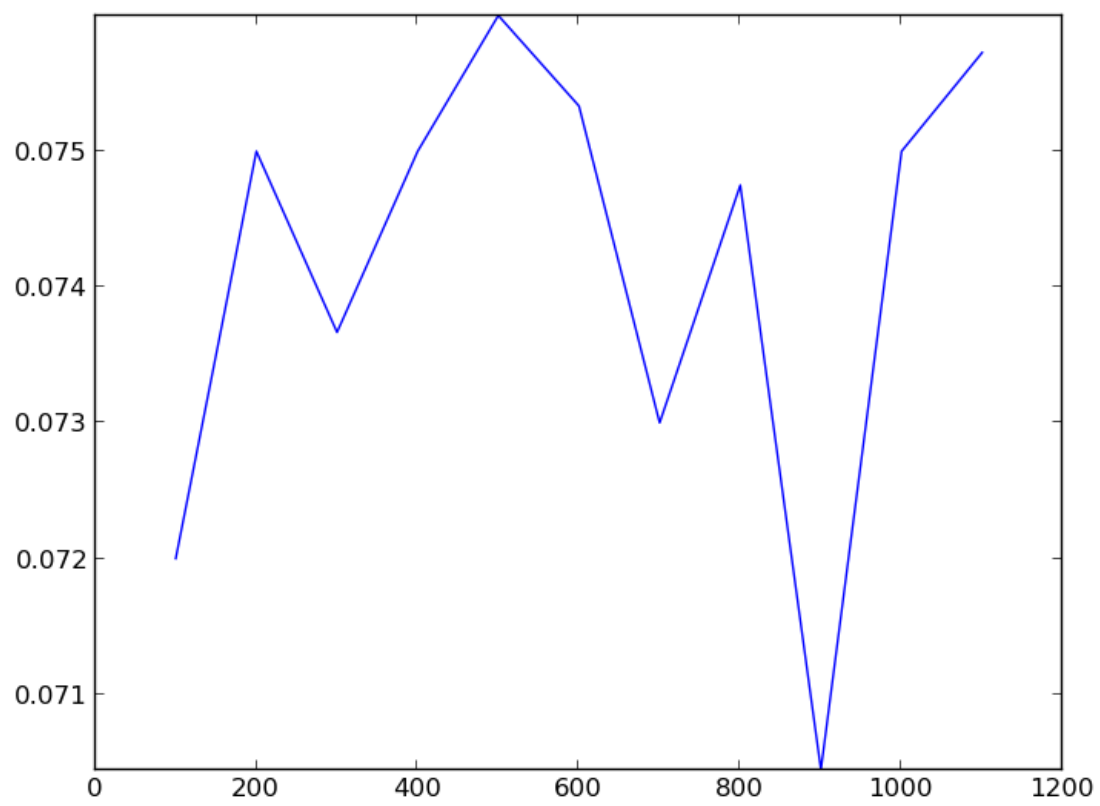
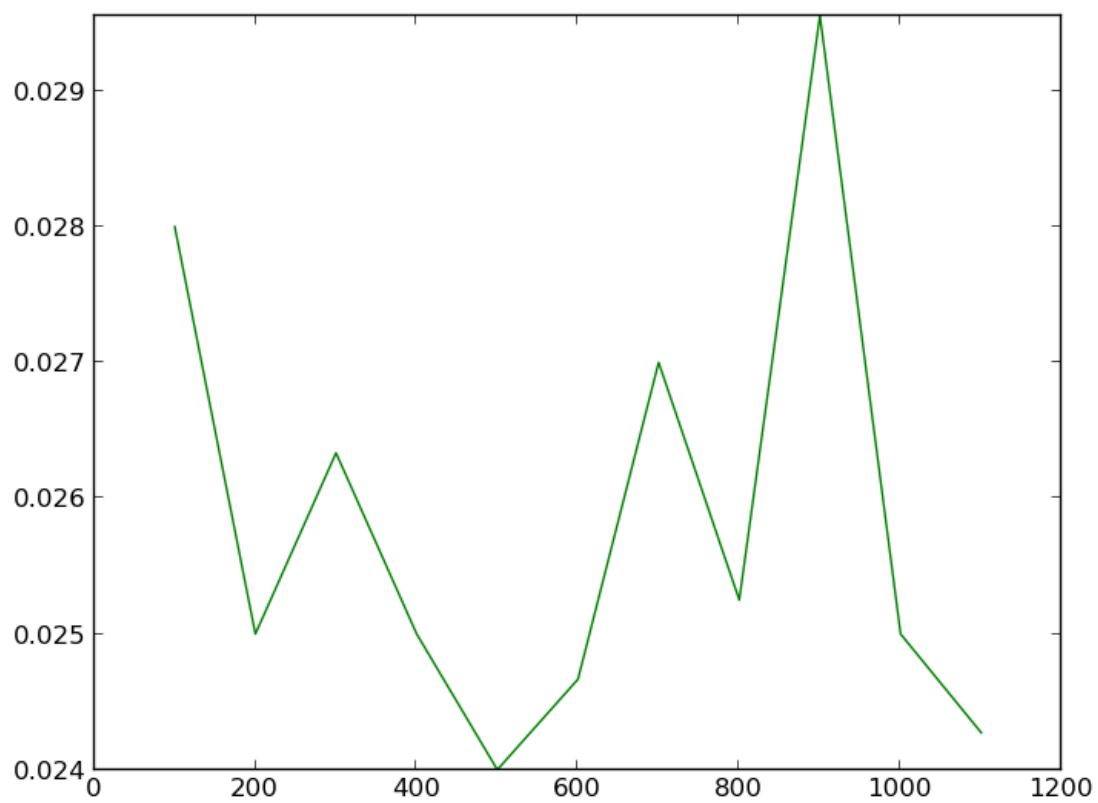


Likelihood Weighting Graphs:

q,-,t,f Graphs:



t,-,q Graph"



2. Yes there was. For the first test case, both converged at around the same number of samples. However, the second test case proved to be problematic for the Rejection Sampling algorithm because earlier, smaller sample sizes through away a lot of samples and thus produced very skewed (close to $\langle 1.0, 0.0 \rangle$ or $\langle 0.0, 1.0 \rangle$) averages. This was shown when I ran the input (q,-,t,f) to test for convergence. The first few outputs were:

0.1 0.0, sample size 100, All values rejected

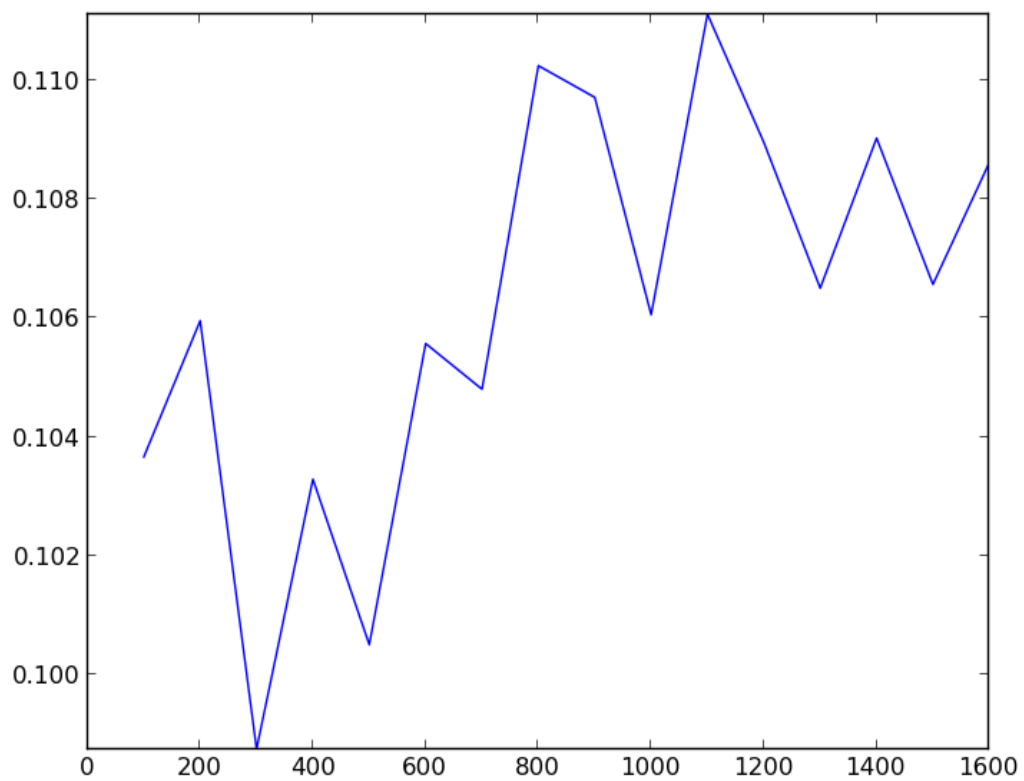
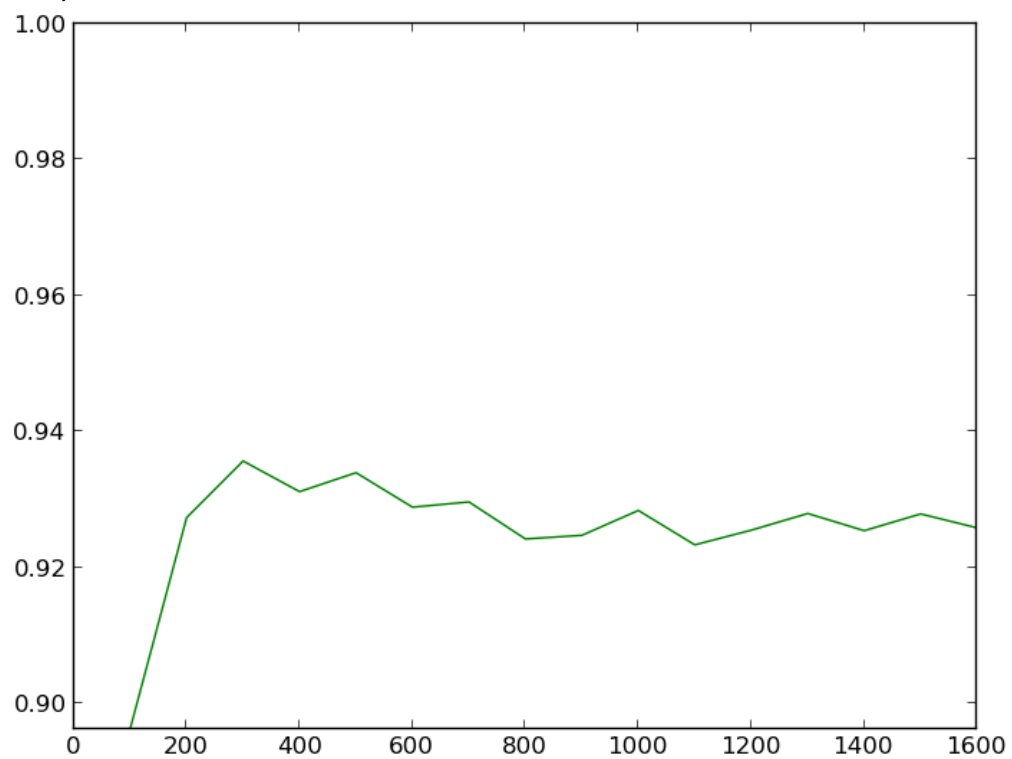
0.1 0.0, sample size 200, All values rejected

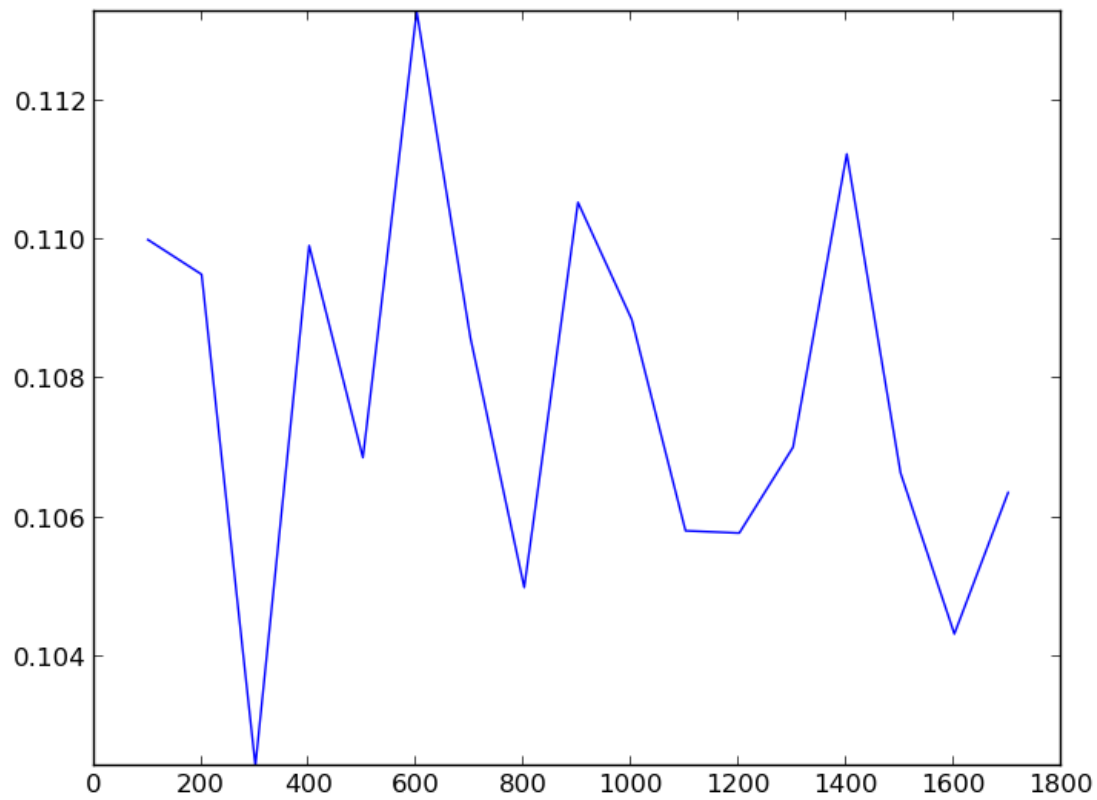
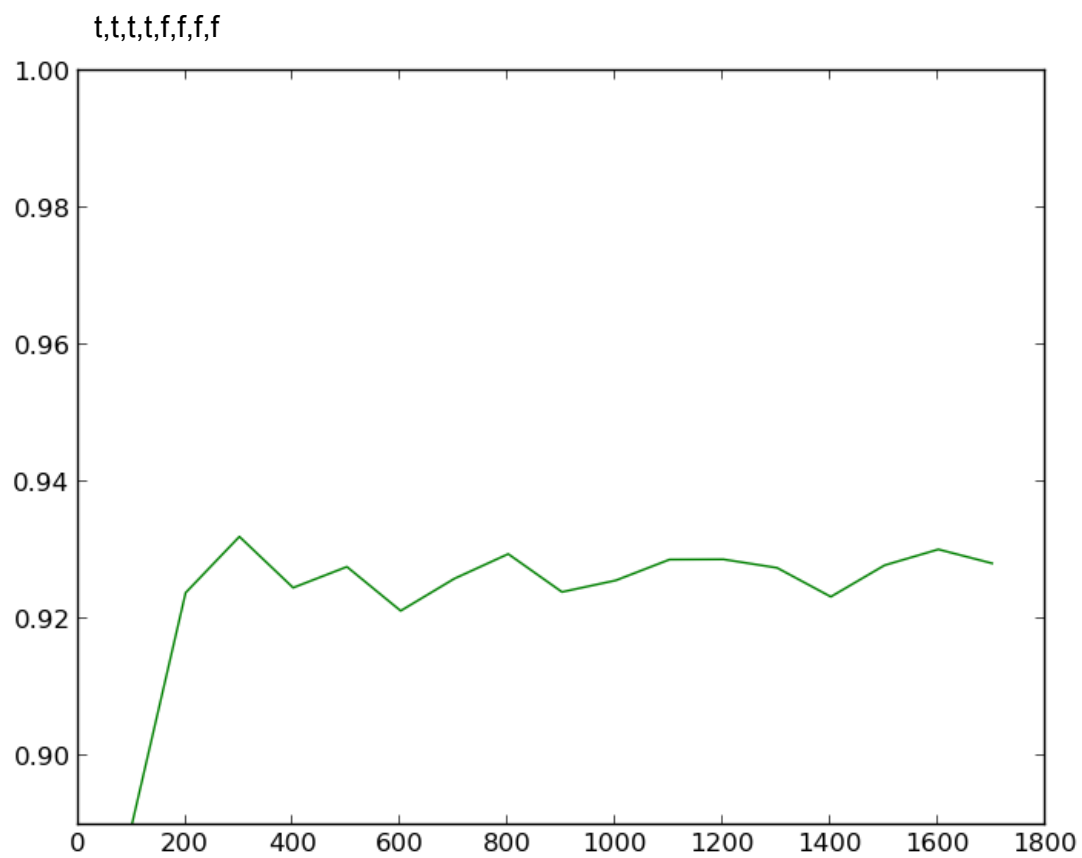
Format (average_true, average_false, sample_size, if values are all rejected)

My windowed, average comparison method, which compared the average of 10 runs for sample size at step t with the averages for steps t-1 down to t-10, and compared the maximum error to an epsilon value, then chose the error for the results with a lot of samples thrown away. Thus Rejection Sampling needed comparisons against averages of runs which through less and less samples were thrown away in order to converge to a given value. This in turn means that it needs an increasing number of samples sizes to converge.

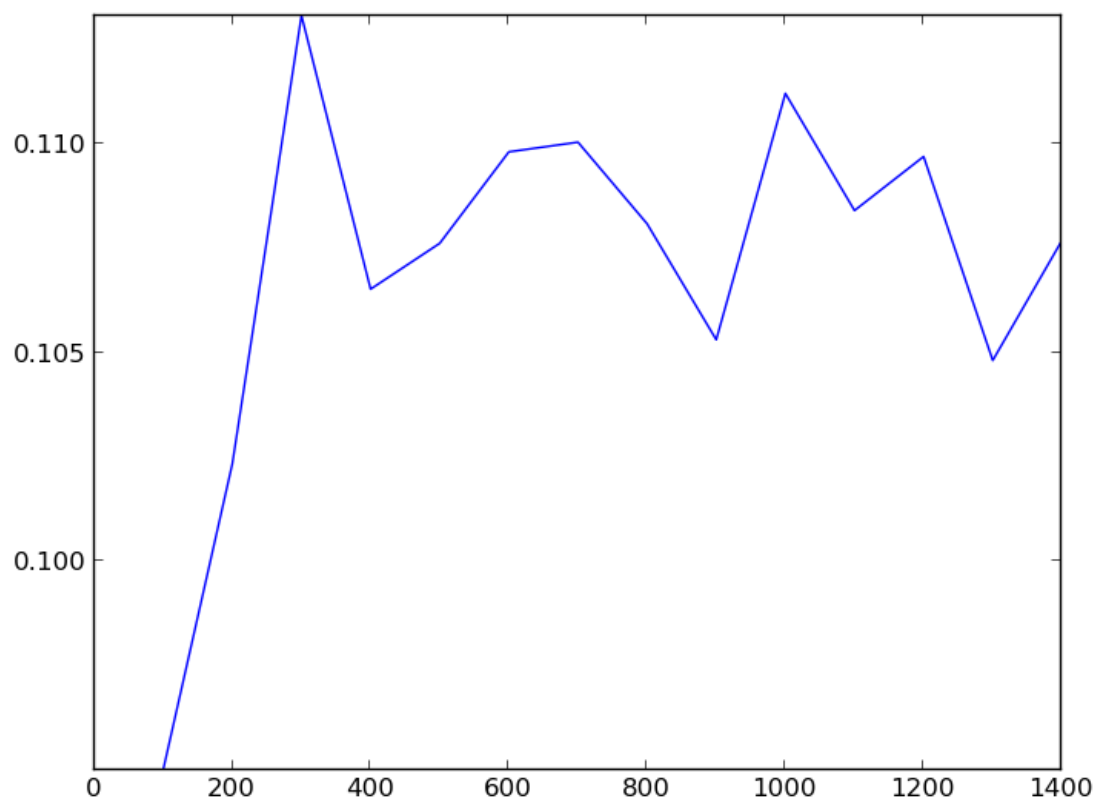
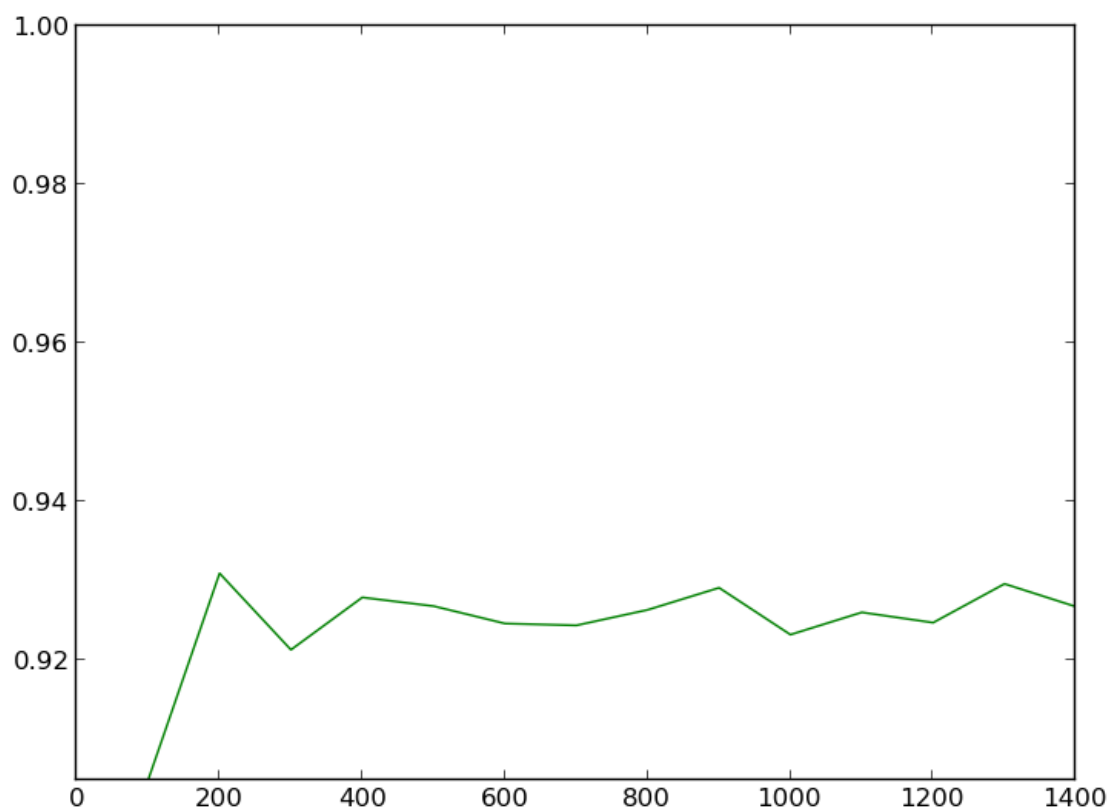
Filtering Writeup:

1. Graphs: t,f,t,f,t,f,t,f





t,t,t,t,t,f



2. For the differences in convergence rate, I initially used $\epsilon = 0.01$ to test for convergence. However, this led about equal rates for all 3 samples, at around 1,200 particles, so I decided to shrink ϵ to 0.005.

Sample input 1 (t,f,t,f,t,f,t,f) performed the best because. This is because at the algorithm resamples the particles more evenly in this input. At every increment of t , we have that either true or false are weighted equally or true outnumbered false by only 1. This means that during the weighting phase, the true particles are weighted either evenly or slightly more than the false particles. Thus the distribution is not extremely skewed to one value during the resampling phase. Resampling with respect to weight has an almost equal likelihood of drawing a true particle as drawing a false particle.

3. The third input does not produce good estimates for smaller number of particles because

the resampling phase is more likely to draw a true particle than a false particle, and thus false particles get thrown away. This is because, like in question 2, during the weighting phase, the true particles outweigh the false particles drastically and this causes the true particles to be more heavily favored during the weight-based probability sampling in the resampling phase.

A possible improvement on particle filtering to be able to handle such inputs properly would be to weigh the likelihood of resampling by the similarity of the current set of particles and the previous set, in the sense that how much does each individual particle differ between sets. This will allow the particle filter to resample less likely during what may be considered a “halted” sequence, wherein a string of observations are seemingly static. This will prevent the constant skewing of the particle values toward the value of the observation in the “halted” sequence.