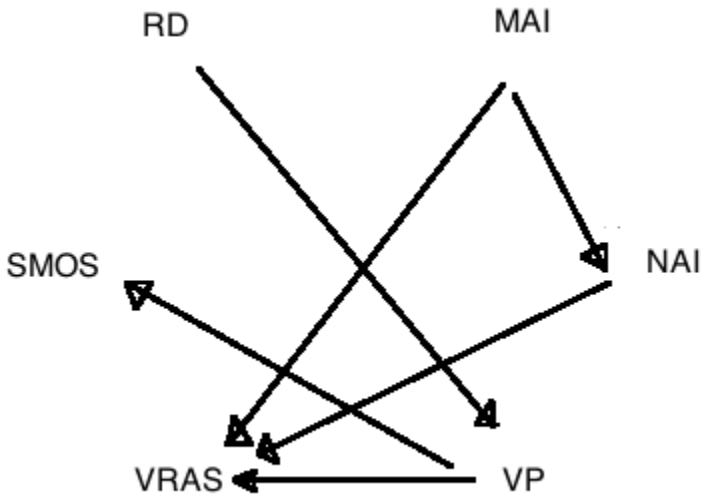
CS181 Assignment 4

Ashok Cutkosky and Tony Feng

April 12, 2013

Problem 1.

(a) According to the graph depicted below, we need to describe the following intermediate probabilities: $P(RD), P(MAI), P(NAI \mid MAI), P(VP \mid RD), P(SMOS \mid VP)$, and $P(VRAS \mid VP)$.



- First, we assume that $P(RD) = 0.2, P(MAI) = 0.5$.
- Next, we assume that $P(VP \mid RD)$ is given by the following table

	RD = 1	RD = 0
P(VP = 1)	0.9	0.0

and $P(SMOS \mid VP)$ is given by the following table

	VP = 1	VP = 0
P(SMOS = 1)	0.9	0.0

and $P(NAI \mid MAI)$ is given by the following table

	MAI = 1	MAI = 0
P(NAI = 1)	0.1	0.9

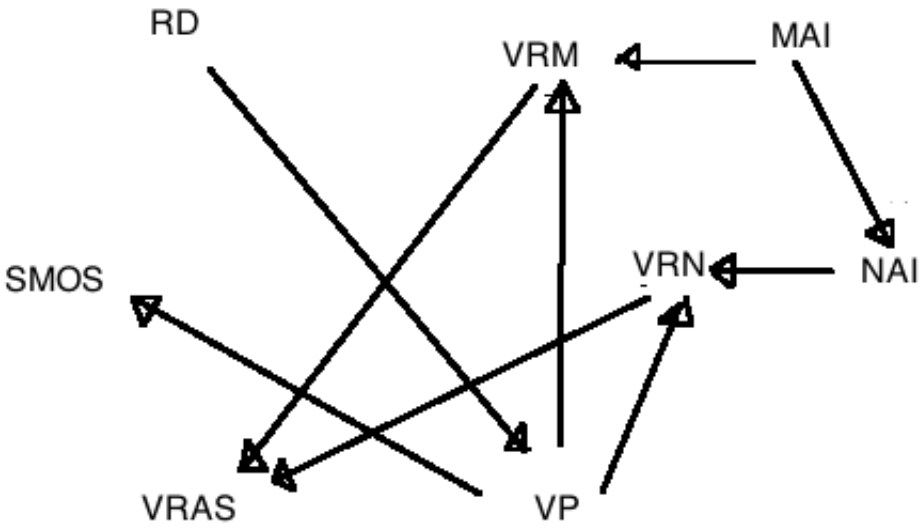
- Finally, we assume that $P(VRAS)$ is given by the following table:

(VP, MAI, NP)	(0,*,*)	(*, 0, 0)	(1,1,0) or (1,0,1)	(1,1,1)
P(VRAS = 1)	0	0	0.7	0.9

(b) Above, we assumed that SMOS (silly messages on screen) was independent of RD (recent download) conditioned on VP (virus present). However, it could be argued that one could have SMOS, e.g. from the ads that tend to inhabit untrusted sites, without actually having a virus. In this case, we should add an edge from RD to SMOS, since we can imagine that RD indicates browsing of the kind of sites that would pop up such silly messages.

We chose not to include this edge for the sake of simplicity, since we expect the majority of silly messages to come from viruses. The problem at hand seems to be that of modeling the virus process, and we can adjust our definition of "silly messages" to be pretty sure that they correspond to viruses.

(c) Here is our new graph.



The things to change are $P(VRM \mid MAI, VP), P(VRN \mid NAI, VP)$, and $P(VRAS \mid ...)$. We are modifying our parametrization slightly so that $P(VRM \mid MAI, VP) = 0.7$ and $P(VRN \mid NAI, VP) = 0.75$, and $VRAS = 1$ if either VRM or $VRN = 1$.

(d) Well, introducing these nodes allows greater flexibility of expression in the modeling process because it allows us to separate the dependencies on the two different antivirus programs. For instance, the graph lets us think about the probability that McAfee detects a virus and the probability that Norton detects a virus separately.

Whether or not this is beneficial to the modeling process depends on our specific goals and situation. It is conceivable that this extra complexity could lead us towards overfitting. Note also that it could already be captured by the parameters of the simpler model, so one could argue that there is not, strictly speaking, more information in this second model.

Problem 2

(a) For graph (a), there are no such variables. For graph (b), the independent variables are F, C. This was determined using the algorithm for detecting d-separation on page 75 of Koller and Friedman, *Probabilistic Graphical Models.* For the first graph, G and D were marked, and none of the vertices are blocked. For the second, G, D, B are marked, and C, F are blocked at I.

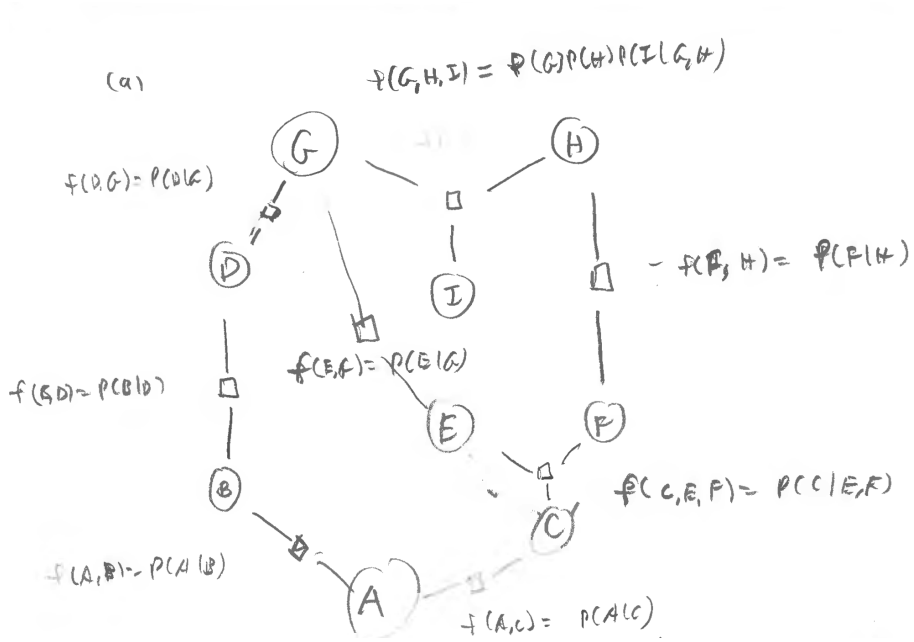
(b) For graph (a), $P(A, B, C, D, E, F, G, H, I)$ is

$$P(G)P(H)P(I \mid G, H)P(D \mid G)P(E \mid G)P(F \mid H)P(B \mid D)P(C \mid E, F)P(A \mid B, C).$$

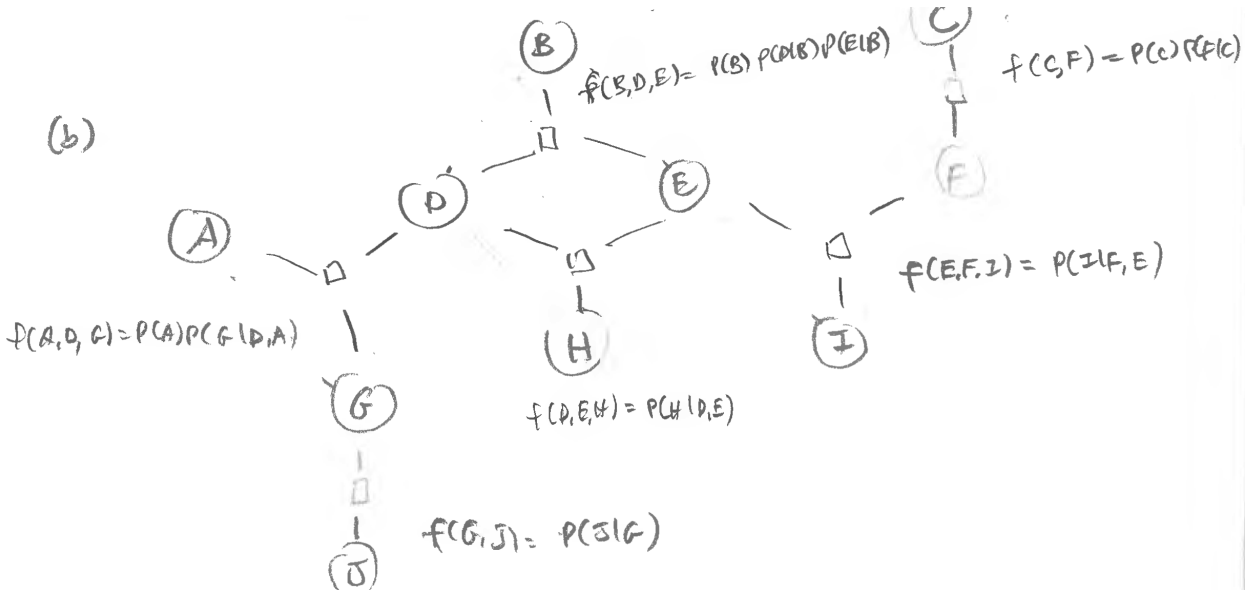
For graph (b), $P(A, B, C, D, E, F, G, H, I)$ is

$$P(B)P(C)P(A)P(D \mid B)P(E \mid B)P(F \mid C)P(G \mid A, D)P(H \mid D, E)P(I \mid E, F)P(J \mid G).$$

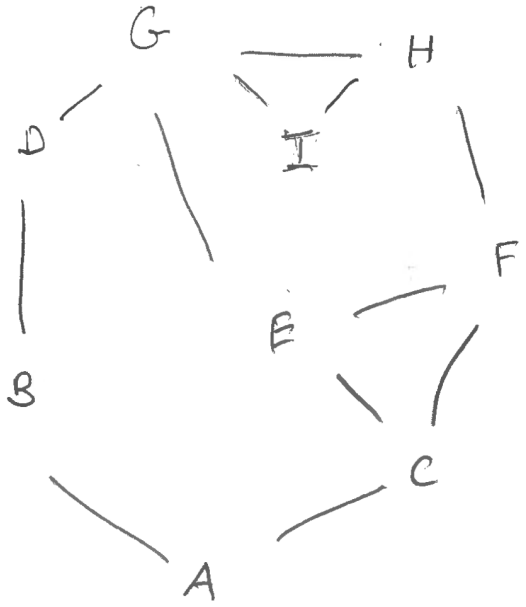
(c) See the image. We have labeled the functions at each square to give the right probability distribution. For graph (a),



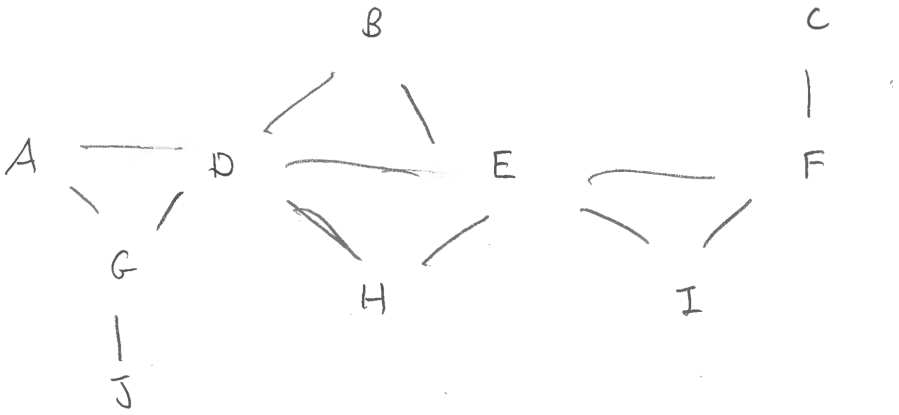
For graph (b),



(d) See the image. Since the Undirected Graphical Models embed fully faithfully into the Factor Graphs, we do not need to label the functions again (they are the same). For graph (a),

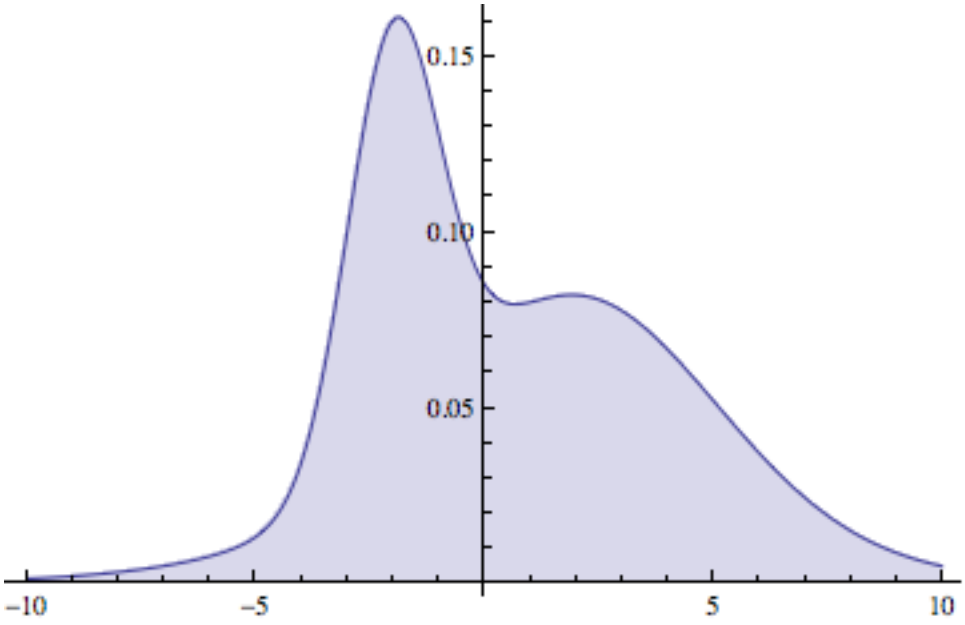


For graph (b),



Problem 3

(a)

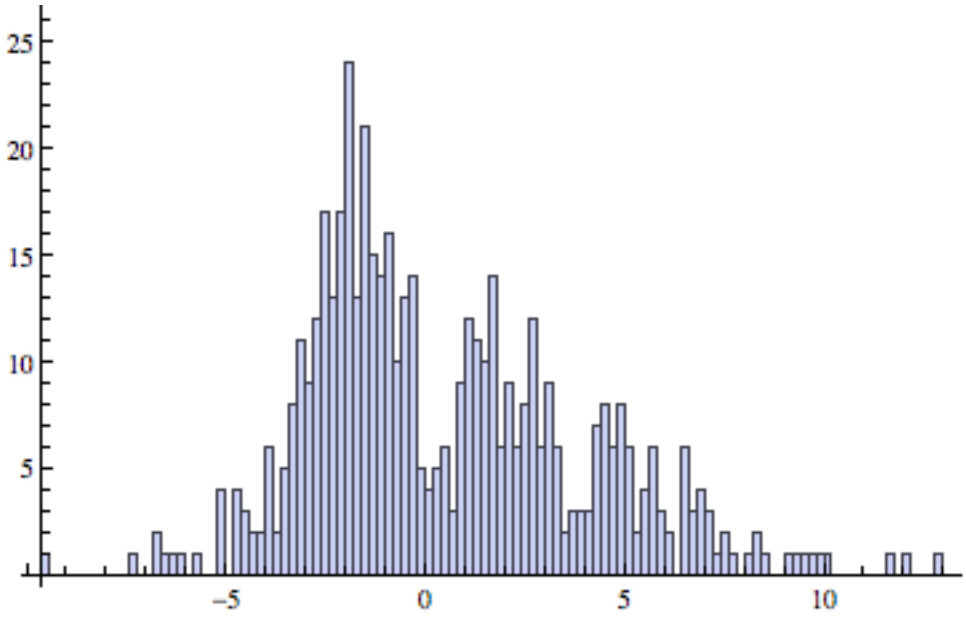


(b) We draw in two steps.

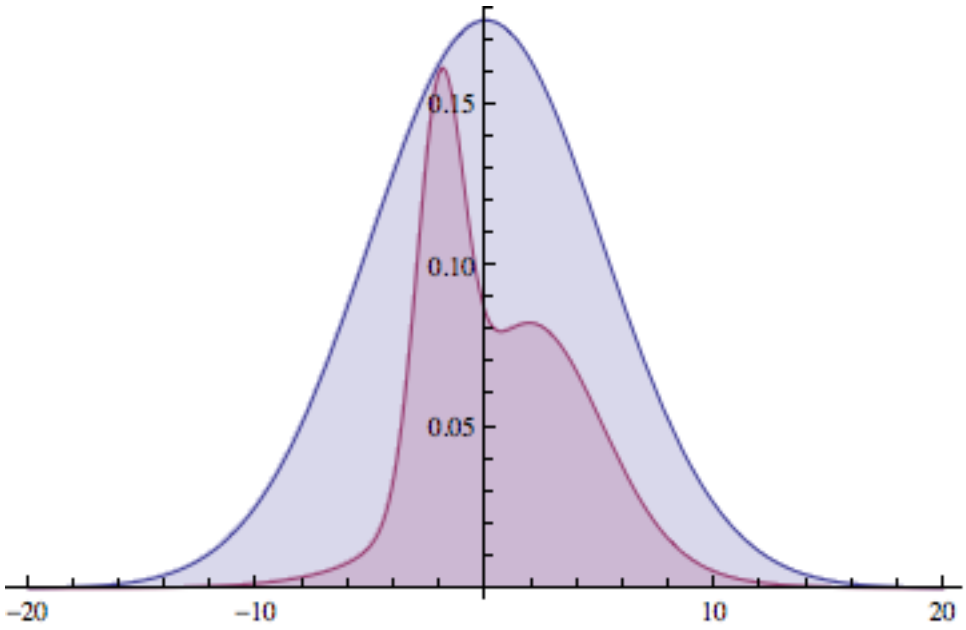
1. First draw randomly from the discrete distribution to determine from which Gaussian to draw: with probability 0.2 we draw from the first, with probability 0.3 we draw from the second, and with probability 0.5 we draw from the third.

2. Second draw from the Gaussian distribution determined in Step 1.

Using this algorithm, we produced the following histogram.



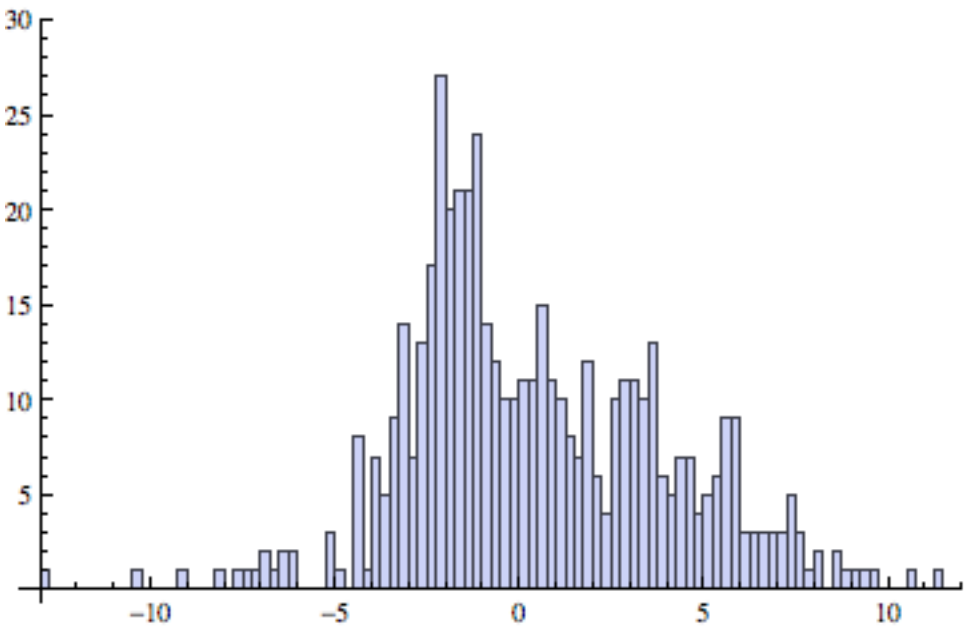
(c) We chose to use the envelope function $g(x) = 2\mathcal{N}(\mu = 0, \sigma = 5.1)$, depicted below.



For let us discuss why it actually fits the condition $g(x) > f(x)$ for all x. A challenge with Gaussian distributions is to bound the tail. Our examples has a greater variance than any of the constituent Gaussians in f, so its tail will dominate their tails for large x (consider the term e^{-x^2/σ^2}, where σ controls the rate of exponential decay).

We wanted something with small tails to mold the small tails of f closely, and another Gaussian was the natural choice. From there, it was a matter of tweaking the parameters.

Note that our envelop has volume 2.25, so we expect a success rate of $1/2.25$. Indeed, we need 1143 trials for 500 samples, and for comparison $1143/2.25 = 508$. Here is the histogram.



(b) See below for the acceptance rates for various standard deviations.

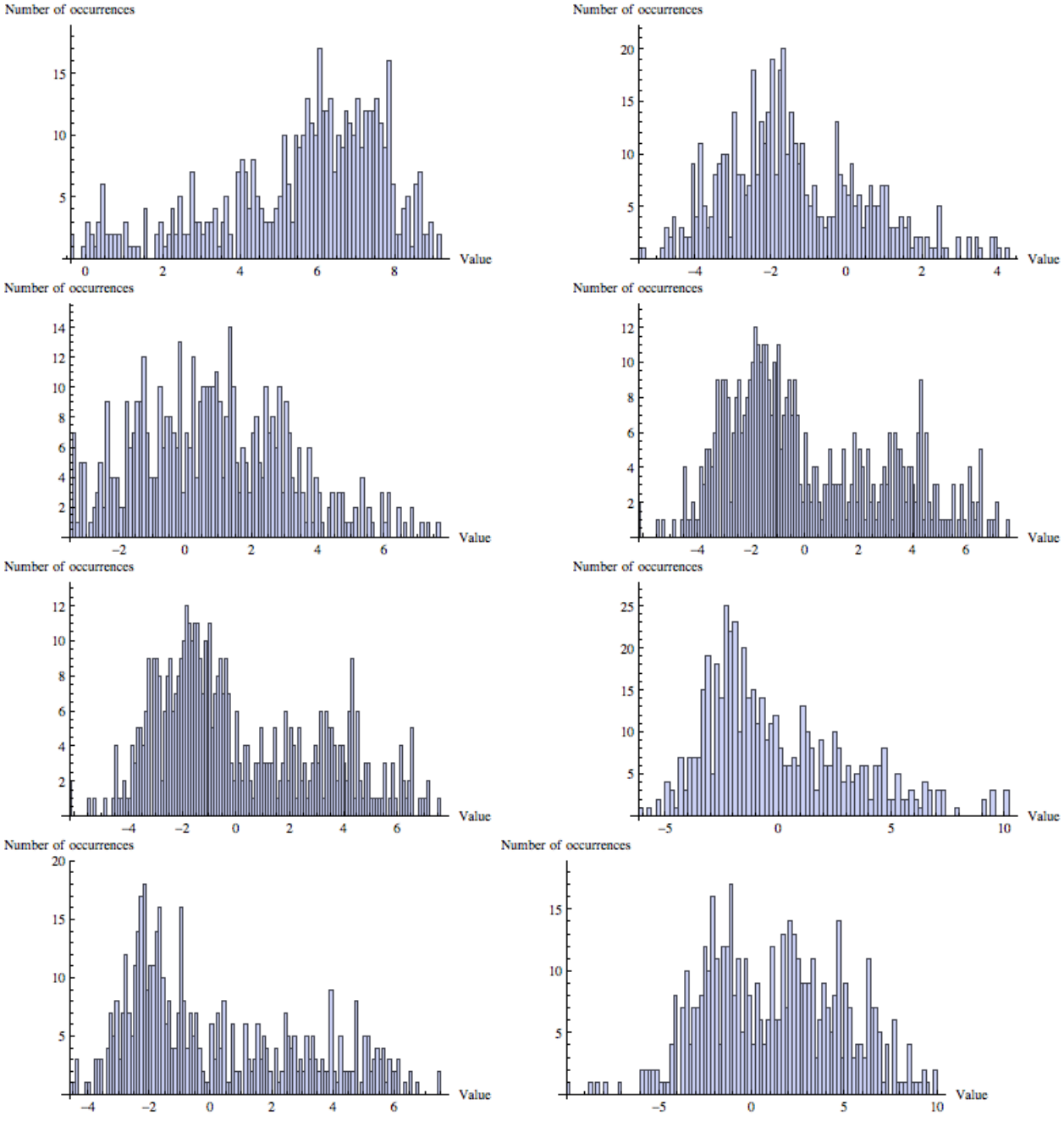
σ	Number accepted
0.25	484
0.50	446
0.75	470
1.00	439
1.25	426
1.50	408
1.75	388
2.00	402
2.25	353

We have depicted, below, the histogram for Gaussians with variance $\sigma = 0.25, 0.5, \ldots, 2.0$.

For small of σ, the algorithm doesn't appear to have converged well to the desired distribution. The "exploration rate" is too low, and the Markov chain tends to move in small steps. The steps are concentrated in high density areas, so the rejection rate is lower.

As σ increases, we see better convergence to the distribution, and a slightly higher rejection rate.

When σ is too large, however, the resolution of the Markov process is wrong again and we expect to see excessively high rejection rates and clumping.


Number of occurrences
Value
Number of occurrences
Value
Number of occurrences
Value
Number of occurrences
Value
Number of occurrences
Value
Number of occurrences
Value
Number of occurrences
Value
Number of occurrences
Value