CS 228 Problem Set 1

Hugh Zhang

March 5, 2017

Problem 1

We want to compute $P(C \mid v_1 \dots v_k)$. For our purposes, our sampler Q is uniform on the possible, so assuming a permutation x' is possible to reach from a state x, then the probabilities are equal. Thus, we can cancel the Qs out.

Acceptance probability is

$$\begin{aligned} & A(c' \mid c, v_1 \dots v_k) \\ = \; & min(1, \frac{P(c' \mid v_1 \dots v_k) Q(c \mid x', v_1 \dots v_k)}{P(c \mid v_1 \dots v_k) Q(c' \mid x, v_1 \dots v_k}) \\ = \; & min(1, \frac{P(c' \mid v_1 \dots v_k)}{P(c \mid v_1 \dots v_k)}) \end{aligned}$$

How do we calculate

$$\frac{P(c' \mid v_1 \dots v_k)}{P(c \mid v_1 \dots v_k)}$$

$$\begin{aligned} & \frac{P(c' \mid v_1 \dots v_k)}{P(c \mid v_1 \dots v_k)} \\ = \; & \frac{P(c', v_1 \dots v_k)}{P(c, v_1 \dots v_k)} \\ = \; & \frac{P(v_1 \dots v_k \mid c') P(c')}{P(v_1 \dots v_k \mid c) P(c)} \end{aligned}$$

We are given $P(v_1 \dots v_k \mid C) = \prod_{v_1 \dots v_k} P(v_i \mid C)$, and although we don't quite know P(c), we are given the assumption that it is uniform a priori, so for our approximation it cancel out.

1.2

The samples are directly from $P(C \mid v_1 \dots v_k)$. Thus,

$$P(C_i = k \mid v_1 \dots v_k) = \frac{\sum_{m=1}^{M} 1(C_i[m] == k)}{M}$$

, or in English, just count the number of times you see it in the sample and divide it by the total number of samples.

1.3

Gibbs sampling does not work. When we sample C, we take two elements C_i and C_j and swap them. If we were to try to Gibbs sample this and try to sample each C_i independently and in order for all i, we would get invalid samples that were not permutations.

Problem 2

From lecture, we have.

$$\begin{aligned} & \log P(y_i \mid x_i, \theta) \\ = \; & \log\left(\frac{1}{Z(x^i,\theta)} \prod_{n \in N} exp(\theta_n f_n(x^i, y_n^i))\right)) \\ = \; & \sum_{n \in N} \theta_n f_n(x^i, y_n^i)) - log(Z(x^i, \theta)) \\ = \; & \sum_{n \in N} \theta_n f_n(x^i, y_n^i)) - log \sum_y exp(\theta_n * f_n(y, x^i)) \end{aligned}$$

Since the likelihood is just the log of the probability summed over all the data, letting M be the number of examples in D, we have

$$g(\theta, D) = (1-\alpha)\ell_{Y|X}(\theta, D) + \alpha \ell_{X|Y}(\theta, D)$$

Where as per above,

$$\begin{aligned} \ell_{Y|X}(\theta, D) &= \sum_i^M (\sum_{n \in N} \theta_n f_n(x_n^i, y_n^i)) - \log \sum_y exp(\theta_n * f_n(y_n, x_n^i))) \\ \ell_{X|Y}(\theta, D) &= \sum_i^M (\sum_{n \in N} \theta_n f_n(x_n^i, y_n^i)) - \log \sum_x exp(\theta_n * f_n(y_n^i, x_n))) \end{aligned}$$

I don't quite want to substitute everything in, but the equation is right there :).

2.2

We want to calculate

$$\begin{aligned} & \frac{\partial}{\partial \theta} g(\theta, D) \\ = \;& (1-\alpha)\frac{\partial}{\partial \theta}\ell_{Y|X}(\theta, D) + \alpha \frac{\partial}{\partial \theta}\ell_{X|Y}(\theta, D) \end{aligned}$$

We calculate

$$\begin{aligned} & \frac{\partial}{\partial \theta}\ell_{Y|X}(\theta, D) \\ = \;& \frac{\partial}{\partial \theta}\sum_{i}^{M}(\sum_{n \in N}\theta_n f_n(x_n^i, y_n^i) - log\sum_{n}\sum_{y} exp(\theta_n * f_n(y_n, x_n^i))) \\ = \;& \sum_{i}^{M}(\sum_{n \in N} f_n(x_n^i, y_n^i) - \frac{\partial}{\partial \theta}\log\sum_{y} exp(\theta_n * f_n(y_n, x_n^i))) \\ = \;& \sum_{i}^{M}(\sum_{n \in N} f_n(x_n^i, y_n^i) - \frac{1}{\sum_y exp(\theta_n * f_n(y_n, x_n^i)))} * \frac{\partial}{\partial \theta}\left(\sum_{y} exp(\theta_n * f_n(y_n, x_n^i))\right)) \\ = \;& \sum_{i}^{M}(\sum_{n \in N} f_n(x_n^i, y_n^i) - \frac{\sum_y exp(\theta_n * f_n(y_n, x_n^i)) * f_n(y_n, x_n^i))}{\sum_y exp(\theta_n * f_n(y_n, x_n^i)))} \\ = \;& \sum_{i}^{M}(\sum_{n \in N} f_n(x_n^i, y_n^i) - \frac{\sum_y exp(\theta_n * f_n(y_n, x_n^i))) * f_n(y_n, x_n^i))}{Z(\theta, x^i)} \\ = \;& \sum_{i}^{M}(f(x^i, y^i) - \sum_{y} P(y, x^i) f(y, x^i)) \\ = \;& M * E_Q[f(x,y)] - M * E_Q[E_{p(y|x)}[f(x^i, y^i)]] \end{aligned}$$

Where Q is the distribution that you got while sampling

$$\begin{aligned} & M(E_Q[f(x,y)] - E_Q[E_{p(y|x)}[f(x,y)]] - \alpha E_Q[f(x,y)] + \alpha E_Q[E_{p(y|x)}[f(x,y)]] \\ & +\alpha E_Q[f(x,y)] - E_Q[E_{p(x|y)}[f(x,y)]]) \\ = \;& M(E_Q[f(x,y)] - E_Q[E_{p(y|x)}[f(x,y)]] + \alpha E_Q[E_{p(y|x)}[f(x,y)]] - \alpha E_Q[E_{p(x|y)}[f(x,y)]]) \end{aligned}$$

Problem 3

E step:

Let $A = |Val(C)|$ and $B_i = |Val(X_i)|$

For each C, we compute

$$
\begin{aligned}
Q(C \mid X) &= \frac{P(X \mid C) * P(C)}{\sum_C P(X \mid C) * P(C)} \\
= & \frac{\theta^0_{x|c} * \theta^0_C}{\sum_C \theta^0_{x|c} * \theta^0_C} \\
= & \frac{\frac{1}{B}\frac{1}{A}}{A * \frac{1}{B}\frac{1}{A}} \\
= & \frac{1}{A}
\end{aligned}
$$

So $Q^1(C \mid X)$ is just uniform over C, as we expect.

M step:

We want to set.

$$
\begin{aligned}
\theta^1 &= \arg\max_\theta \sum_D \sum_C Q^1(C \mid X) \log P(C, X, \theta) \\
= & \arg\max_\theta \sum_D \sum_C \frac{1}{A} \log \theta_C \theta_{x|c} \\
= & \arg\max_\theta (\sum_D \sum_C \log \theta_C + \sum_D \sum_C \log \theta_{x|c})
\end{aligned}
$$

$\frac{1}{A}$ has no bearing on maximizing the parameters. Another way of putting the above is that C has to be uniform at the end, because we have no information for which way to bias C. Thus, θ_C remains constant. According to the lecture notes, the proper way to maximize $\log \theta_{x|c}$ is to set it to the weighted observed counts in the data.

$$
\begin{aligned}
\theta_C &= \frac{1}{A} \\
\theta_{x|c} &= \frac{\sum_D P(c, x)}{\sum_D \sum_x P(c, x)} = \frac{cnt(x)}{M}
\end{aligned}
$$

Since C gives us no information, we it is essentially unifrom MLE and where cnt(x) is how many times x appears in the training data.

To show that it converges there and no longer changes, we hallucinate C again, and it will be uniform since we have no other information on it.

$$
\begin{aligned}
Q(C \mid X) &= \frac{P(X \mid C) * P(C)}{\sum_C P(X \mid C) * P(C)} \\
&= \frac{\theta^0_{x|c} * \theta^0_C}{\sum_C \theta^0_{x|c} * \theta^0_C} \\
&= \frac{\frac{1}{B}\frac{1}{A}}{A * \frac{1}{B}\frac{1}{A}} \\
&= \frac{1}{A}
\end{aligned}
$$

And since $P(C \mid X)$ is as in step 1, nothing will update for the parameters, so this is the final distribution.

Problem 4

Let A denote the set of variables not in either Y or Y's markov blanket. I'm going to split up $Y_M(i,j)$ to include the 2-4 neighboring Y variables, and separately include x_{ij}, which would ordinarily be in Y's markov blanket. The general gist is that everything in A cancels out.

$$
\begin{aligned}
P(y_{ij} \mid y_{M(i,j)}) &= \frac{P(y_{ij}, y_{M(i,j)})}{P(y_{M(i,j)}))} \\
&= \frac{\sum_A P(y_{ij}, y_{M(i,j)}, A)}{\sum_A \sum_{y_{ij}} P(y_{ij} y_{M(i,j)}), A)} \\
&= \frac{\sum_A exp(\eta \sum_N \sum_M y_{ij} x_{ij} + \beta \sum_{edges} y_{ij} y_{i'j'})}{\sum_A \sum_{y_{ij}} exp(\eta \sum_N \sum_M y_{ij} x_{ij} + \beta \sum_{edges} y_{ij} y_{i'j'})} \\
&= \frac{exp(\eta y_{ij} x_{ij} + \beta \sum_{Y_{M(i,j)}} y_{ij} y_{i'j'}) * \sum_A exp(\eta \sum_A y_a x_a + \beta \sum_{restofedges} y_{ij} y_{i'j'})}{\sum_{y_{ij}} exp(\eta y_{ij} x_{ij} + \beta \sum_{Y_{M(i,j)}} y_{ij} y_{i'j'}) * \sum_A exp(\eta \sum_A y_a x_a + \beta \sum_{restofedges} y_{ij} y_{i'j'})} \\
&= \frac{exp(\eta y_{ij} x_{ij} + \beta \sum_{Y_{M(i,j)}} y_{ij} y_{i'j'})}{\sum_{y_{ij}} exp(\eta y_{ij} x_{ij} + \beta \sum_{Y_{M(i,j)}} y_{ij} y_{i'j'})} \\
&= \frac{exp(\eta y_{ij} x_{ij} + \beta \sum_{Y_{M(i,j)}} y_{ij} y_{i'j'})}{exp(\eta x_{ij} + \beta \sum_{y_{i'j'} \in Y_{M(i,j)}} y_{i'j'}) + exp(-\eta x_{ij} - \beta \sum_{y_{i'j'} \in Y_{M(i,j)}} y_{i'j'})}
\end{aligned}
$$

Thus,

$$
\begin{aligned}
&P(y_{ij} = 1 \mid Y_{M(i,j)}, x_{ij}) \\
&= \frac{exp(\eta x_{ij} + \beta \sum_{Y_{M(i,j)}} y_{i'j'})}{exp(\eta x_{ij} + \beta \sum_{y_{i'j'} \in Y_{M(i,j)}} y_{i'j'}) + exp(-\eta x_{ij} - \beta \sum_{y_{i'j'} \in Y_{M(i,j)}} y_{i'j'})} \\
&= \frac{1}{1 + exp(-2\eta x_{ij} - 2\beta \sum_{y_{i'j'} \in Y_{M(i,j)}} y_{i'j'})}
\end{aligned}
$$

Which is a sigmoid function.

4.2

You are given the X's and randomly initialize the Ys. Then, go through the Y's in order and draw from the $P(y_{ij} = 1 \mid Y_{M(i,j)}, x_{ij})$ that we gave above (using the newest values of Y that we just got if necessary). When you have gone through all the Y's you have gone through one ”iteration” of Gibbs Sampling. After burning through the burn in samples, Gibbs Sampling guarantees that we converge to $P(Y \mid X)$ eventually because this is clearly ergodic.

Irreducible: You can get to any Y with sufficient luck because any sample is possible.

Aperiodic: You can hit any Y in one ”iteration” of Gibbs if you sample the right pixel at each step, since each pixel has a non zero chance of hitting 0/1 each sample.

4.3

All three converge to the same approximate energy: -214000. The chains eventually end up mixing for all three starts, which indicates that our Gibbs sampling is not really getting stuck and is doing the right job. The burn in period for the same and the random seems sufficient, since the burn in samples seem mixed in themselves before we start the real samples. With the logneg graph, it is less clear, since the graph hasn't clearly started mixing yet and the beginning of our sample might be not quite $P(Y \mid X)$. A touch longer burn in might be safer.

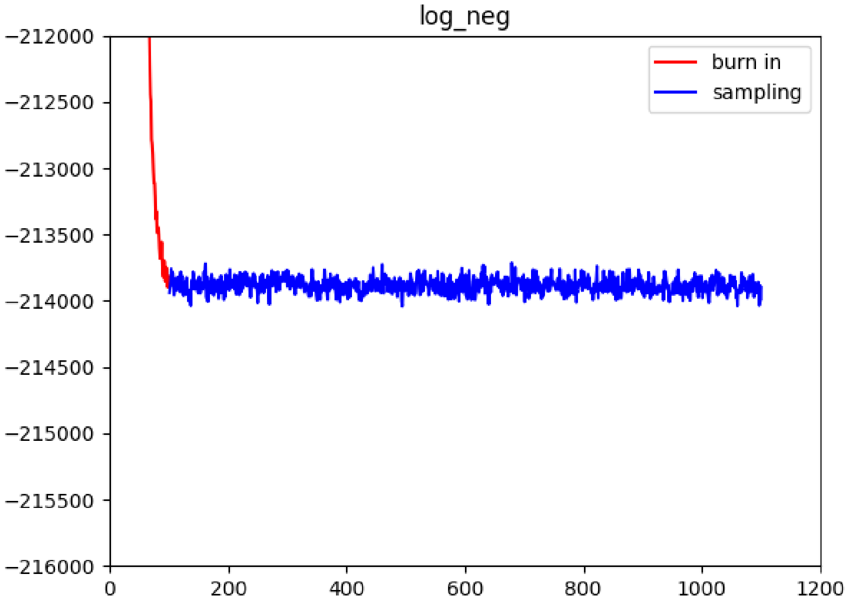

Figure 1: Log Neg

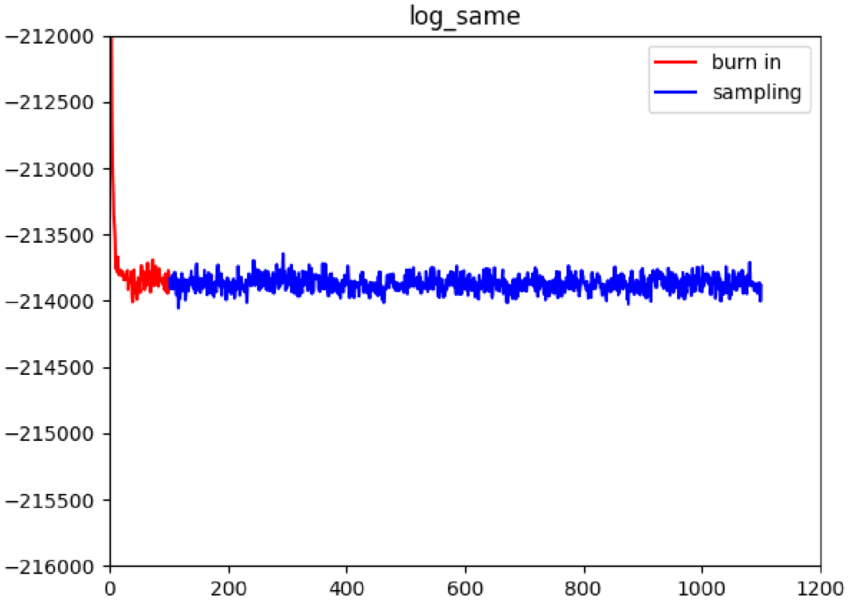

Figure 2: Log Same

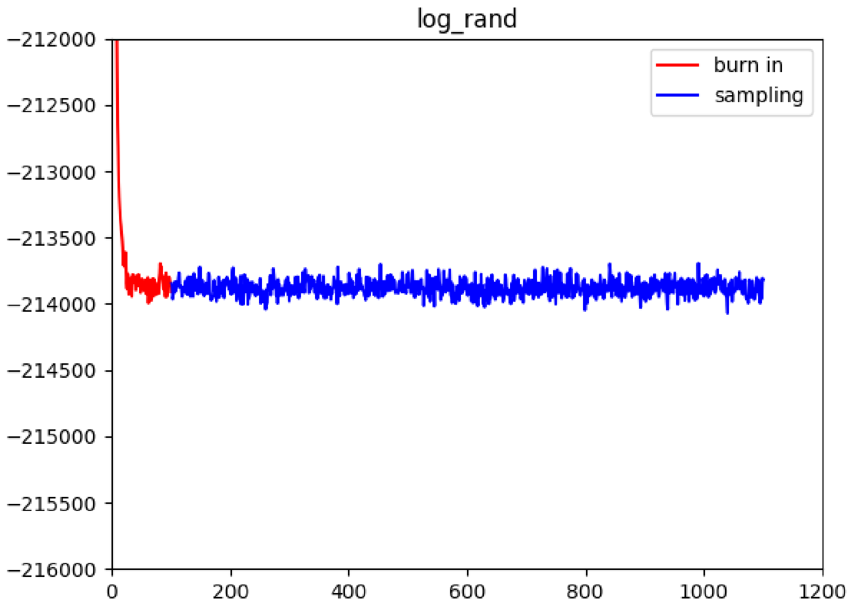

Figure 3: Log Rand

4.4

Correct pixels for 10: 84454 Correct pixels for 20: 84072 Total: 84960

Accuracy 10: 99.4 or .6 error Accuracy 20: 98.95 or 1.05 error

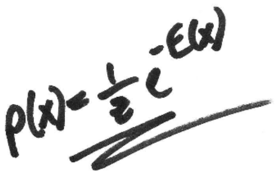

Figure 4: Original

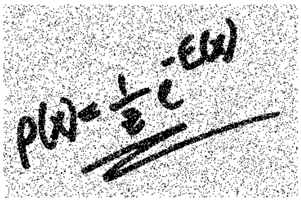

Figure 5: Noisy 10

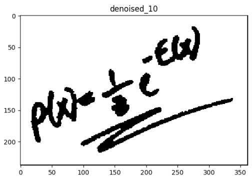

Figure 6: Denoised

4.5

Correct pixels for 10: 84340 Correct pixels for 20: 82995 Total: 84960

Accuracy 10: 99.27 or 0.73 error Accuracy 20: 97.69 or 2.31 error

Caveats: The accuracies vary slightly based on what you do for ties and what order you iterate through the photo. But in general, it performs similarly but

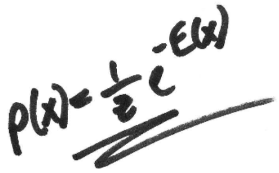

Figure 7: Original

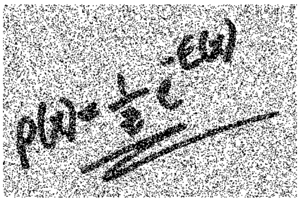

Figure 8: Noisy 20

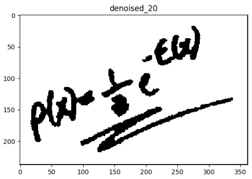

Figure 9: Denoised

slightly worse to Gibbs sampling. This is because, our version of Gibbs sampling is doing a very similar algorithm with η and β both being 1, you are essentailly looking at your neighbors and seeing which direction they want you to go. The difference is that Gibbs sampling is probabilistic so it explores more and doesn't get suck on a suboptimal photo (although notice that it does require more iterations 1100 to 30)

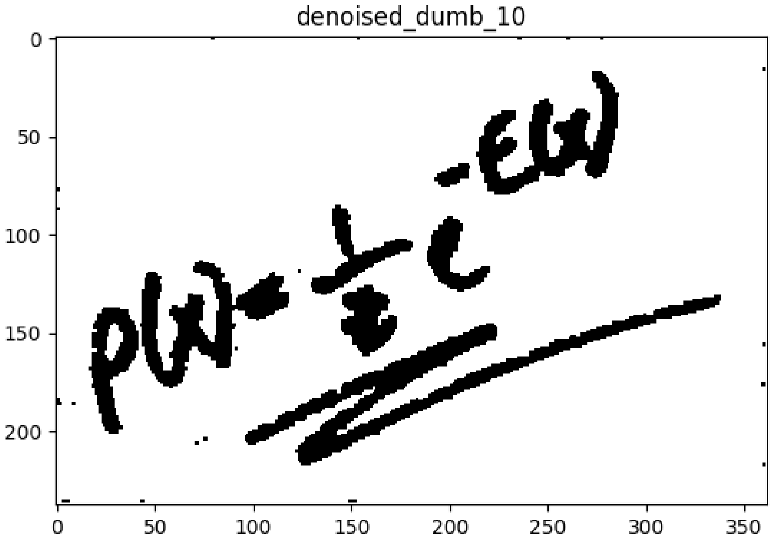

Figure 10: Bad Algo Restored 10

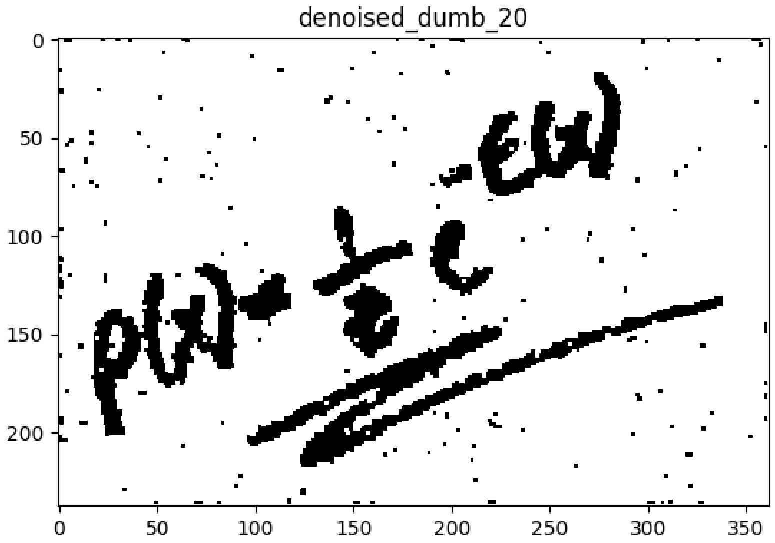

Figure 11: Bad Algo Restored 20

4.6

Both normalish distributions. Noisy 10 has slightly less pixels on average (703 ish?) compared to Noisy 20 (718 ish?), which is better, because the original image has 680 pixels in that rectangle which is lower than either of them

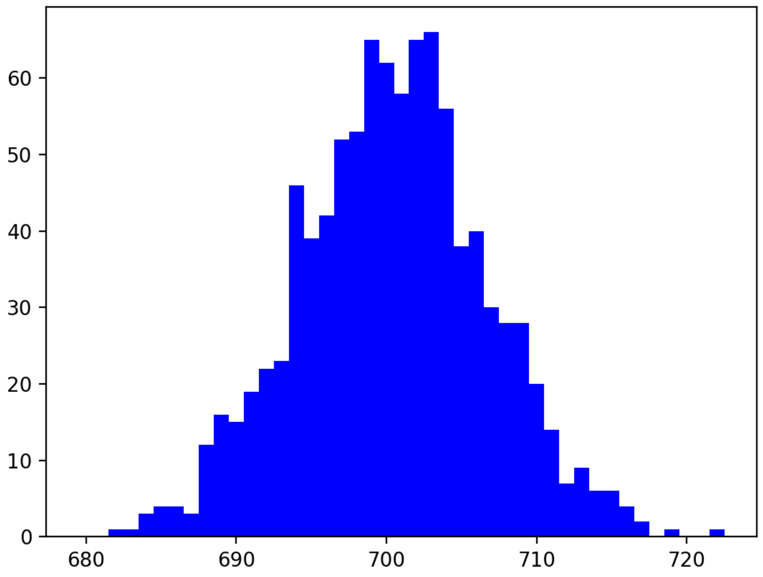

Figure 12: Noisy 10 picture hisogram

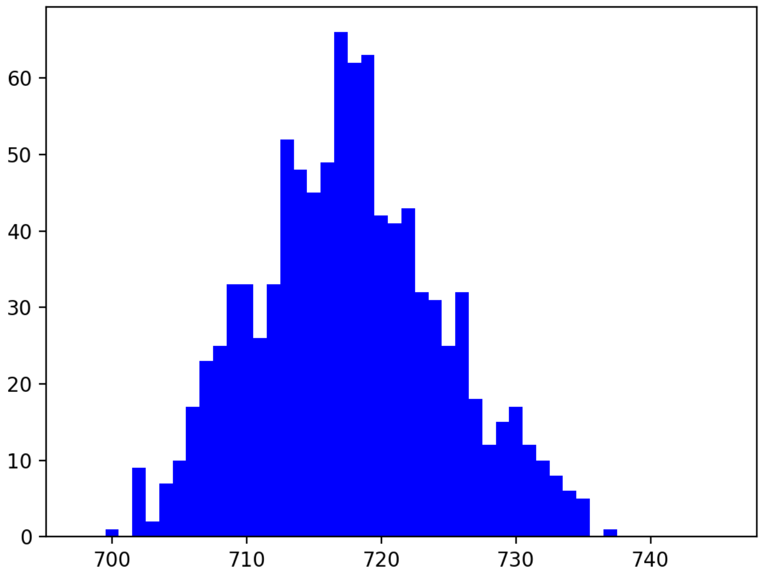

Figure 13: Noisy 20 picture hisogram