Final project

Yingxue Su

April 2019

1 Generate data and sample the data

We have a square $[0,1]\times[0,1]$. First, we discrete this square with 10000 points $(a\times 0.01, b\times 0.01)$ for $a,b=1,2,...,100$. Then we generate 4 random numbers between 0 and 1 to separate the square to 4 areas in figure 1.

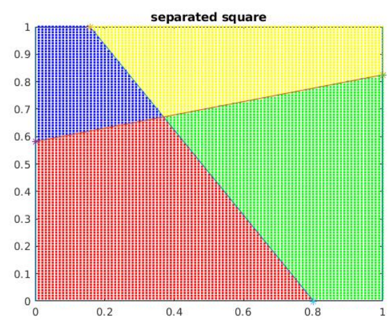

Figure 1: separated square with 10000 points

Then we sample 2000 points from these 10000 points in figure 2. So the red

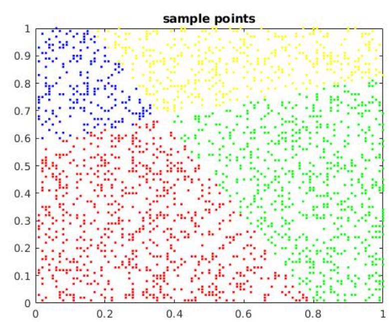

Figure 2: 2000 sample points

area (class 1) has 739 points, the blue area (class 2) has 203 points, the green area (class 3) has 660 points, the yellow area (class 4) has 398 points. Define the cost function as following:

$$cost = \sum_{j=1}^{4} \sum_{i|c_i=j} ||(a_i, b_i) - center_j|| \times \frac{1}{num_j}$$

where (a_i, b_i) is the point and c_i is its class, num_j is the size of class j, $center_j = \frac{1}{num_j} \sum_{i|c_i=j} (a_i, b_i)$. For figure 2, $cost_{ini} = 0.8121$.

2 Boltzmann Machine Simulation

We want to keep (a_i, b_i) same for $i = 1, 2, ..., 2000$, but hide the class c_i for every point. And assign random class to every point for the initial class. Then for every sweep, we visit all 2000 points. When we visit (a_i, b_i), if the currant class of (a_i, b_i) is 2, we attempt to change its class to 1,3,4 with probability 1/3 for each value. For example, after change, the new class is 4. We calculate the change of cost

$$\Delta cost = New \ \ cost - Old \ \ cost.$$

And $\Delta cost$ is only related to points that belong to class 2 and class 4. If < 0, we keep the change and visit next point. If $\Delta cost > 0$, we accept the change with probability $p = exp(-\Delta cost/T)$, where $T = 0.9^{sweep}$ or $T = 0.99^{sweep}$. Let $T = 0.9^{sweep}$ we do this for 500 sweeps, and every time we visit 100 points, we calculate $cost$ and plot it in figure 3. We can see that the $cost$ stabilize for a while then decrease fast and stabilize 0.41 which is less than $cost_{ini}$. The cost function does not decrease right away was because for the 15 sweeps T was still not so small, so p was very close to one for $\Delta cost > 0$. And every sweep takes around 1.2 seconds.

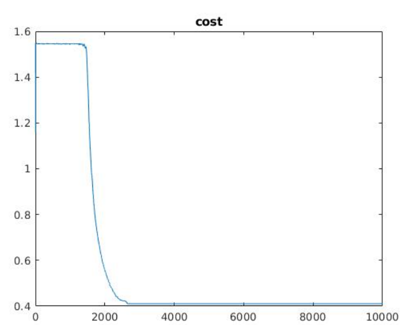

Figure 3: cost with $T = 0.9^{sweep}$

We plot the sample points with the classes assigned by this simulation in figure 4. We can see that most points are in class 4 (yellow) now. Only 6 points are in other 3 classes with 2 points for each. If we calculate the cost when the whole square is only one class, $cost \approx 0.38$. So the minimum the Boltzmann machine finds happens when the whole square is covered by mostly one class.

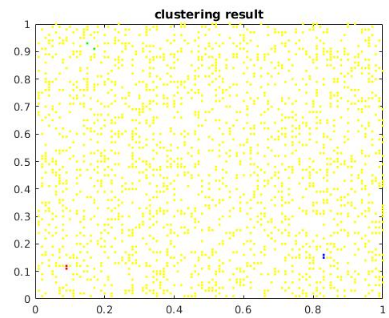

Figure 4: clustering result

If we run the simulation again with $T = 0.99^{sweep}$, and plot the cost function for both $T = 0.9^{sweep}$ and $T = 0.99^{sweep}$ in figure 5. We can see that when $T = 0.99^{sweep}$, it takes longer for cost function to reduce. It is because when $T = 0.99^{sweep}$, it takes longer for T to become small. So for the first hundreds sweeps, when $\Delta cost > 0$, we accept the change with big probability.

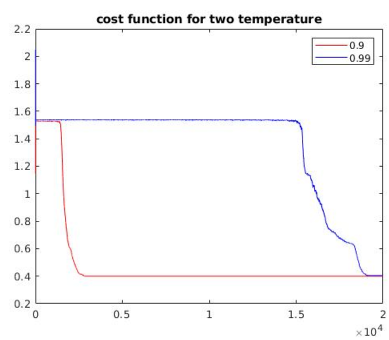

Figure 5: cost function with $T = 0.99^{sweep}$ and $T = 0.9^{sweep}$

3 Add penalty to cost function to control the number of classes

Since in section 2, the clustering result end up with one class, we will add a penalty part to control the number of classes. Same as before, we discrete the square with 10000 points and separate it to 4 areas (figure 6). Then we take 2000 points for a random sample in figure 7.

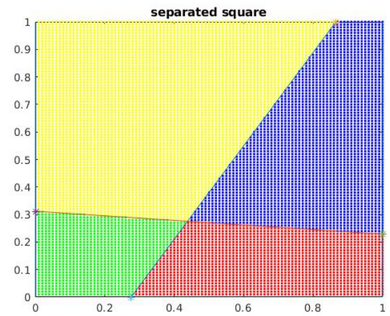

Figure 6: separated square with 10000 points

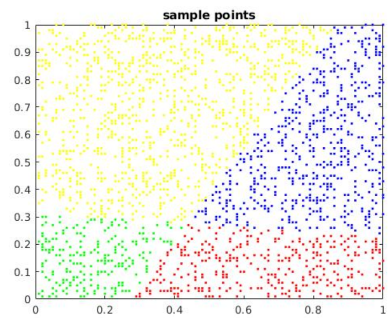

Figure 7: 2000 sample points

Then we redefine the cost function as following:

$$cost = \sum_{j=1}^{4} \sum_{i|c_i=j} ||(a_i, b_i) - center_j|| \times \frac{1}{num_j} + \frac{1}{4}\sum_{j=1}^{4}(a - s_j)^+ \times 10$$

where $s_j = num_j/2000$ and a is the lowest percentage of 4 classes of the sample before simulation. And $(m)^+$ means the positive part of m. Then the cost of the sample for figure 7 is $cost_{ini} = 0.80$. If the 2000 points are all in one class, the cost is $cost = 1.16$ which is even bigger than $cost_{ini}$. So this time, the clustering result will not end up with only one class. Let $T = 0.9^{sweep}$, and run for 500 sweeps. Plot the energy function and the clustering result in figure 8 and 9 respectively. The cost function stabilizes around 0.70 which is less than $cost_{ini}$. However, the clustering result is still not satisfying, which is understandable. Because we did not give enough information in the cost function.

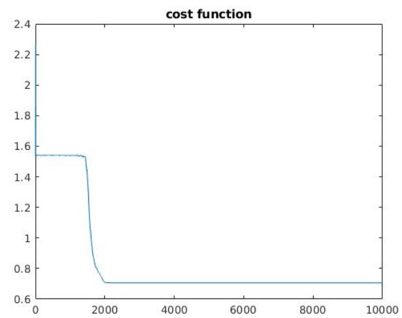

Figure 8: cost with $T = 0.9^{sweep}$

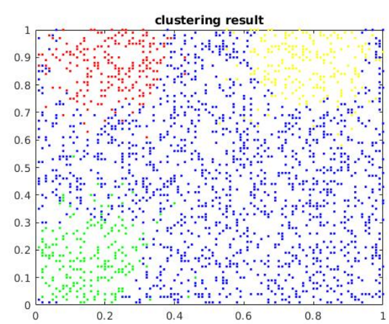

Figure 9: clustering result