Pattern Classification and Recognition
ECE 681

Spring 2019
Homework #5: Nonlinear Dimensionality Reduction (t-SNE)
(In-Class Assignment)

Due: 5:00 PM, Thursday, March 7, 2019
Grace Period Concludes: 11:30 PM, Tuesday, March 19, 2019

This homework assignment is worth **110 points**.
Each problem is worth some multiple of 10 points, and will be scored on the below letter scale.
The letter grades B through D may be modified by + (+3%) and A through D may be modified by a - (-3%).

A+ = 100%: Exceeds expectations, and no issues identified
A = 95%: Meets expectations, and (perhaps) minor/subtle issues
B = 85%: Issues that need to be addressed
C = 75%: Significant issues that must be addressed
D = 65%: Major issues, but with noticeable perceived effort
F = 50%: Major issues, and insufficient perceived effort
Z = 30%: Minimal perceived effort
N = 0%: Missing, or no (or virtually no) perceived effort

Your homework is not considered submitted until both components (**one self-contained pdf file** and your code) have been submitted. Please do not include a print-out of your code in the pdf file.

You should strive to submit this assignment by the due date/time. The grace period is intended to afford an opportunity to, for example, work through technical glitches, update your submission if you realize after the due date that you submitted the wrong file or would prefer to answer a question differently, and provide some flexibility to manage your workload as it ebbs and flows during the semester.

Visualizing Flip Sequences from Pennies and Quarters

We are re-visiting the (impossible) machine learning problem we considered when exploring the curse of dimensionality: Distinguishing pennies from quarters based solely on the sequence of Heads/Tails flips they produce. The coin is in another room, where someone else will flip it some number of times and tell us the result. That is, the observations we get in order to classify the coin as either a penny or a quarter are a series of coin flip results, heads or tails.[1]

The data for this problem is the sequence of Heads/Tails flips from each coin. If P pennies and Q quarters ($N = P + Q$) were each flipped D times the resulting dataset might look like this:

$$X = \begin{bmatrix} \text{T} & \text{H} & \dots & \text{T} \\ \text{H} & \text{T} & \dots & \text{T} \\ \vdots & \vdots & \vdots & \vdots \\ \text{H} & \text{H} & \dots & \text{H} \end{bmatrix} \qquad Y = \begin{bmatrix} \text{q} \\ \text{q} \\ \vdots \\ \text{p} \end{bmatrix}$$

where X contains the results of the coin flips and is of size $N \times D$ (coins × flips), and Y contains the truth as to which coin was flipped and is of size $N \times 1$ (coins × 1). The "data" (X) are the coin flip results: heads (H) or tails (T). The "truth" or "labels" (Y) are either quarter (q) or penny (p).

We can encode these data and associated labels by assigning 0 to represent tails and 1 to represent heads, and 0 to indicate the coin flipped was a penny and 1 to indicate the coin flipped was a quarter. With this encoding, the data and labels are:

$$X = \begin{bmatrix} 0 & 1 & \dots & 0 \\ 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix} \qquad Y = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 0 \end{bmatrix}$$

We are going to use t-distributed stochastic neighbor embedding (t-SNE) to visualize the data (D-dimensional sequences of coin flips), and investigate the effects of the number of observations and the number of dimensions on the resulting visualization. There are t-SNE implementations available for both Matlab and Python; I strongly encourage you to use one of these implementations over writing your own.

[1] We will assume all the coins are perfectly fair, that is there is no dynamical bias in these coin flips (*i.e.*, $p(\text{H}) = 0.5$ and $p(\text{T}) = 0.5$). See "Dynamical Bias in the Coin Toss" by Persi Diaconis, Susan Holmes, and Richard Montgomery for a discussion of dynamical bias in coin flips (posted to Sakai under Syllabus and Additional Resources → Interesting Articles).

(20) 1. Consider first the case where each coin is flipped twice ($D = 2$).

(a) Simulate sequences of coin flips for $P = 5$ pennies and $Q = 5$ quarters. Visualize the data using t-SNE, clearly denoting the data points that are associated with common coin flip sequences, *i.e.*, use color and or marker type to encode which data points are due to HH, which are due to HT, which are due to TH, and which are due to TT.

Figure: P and Q are not seperated

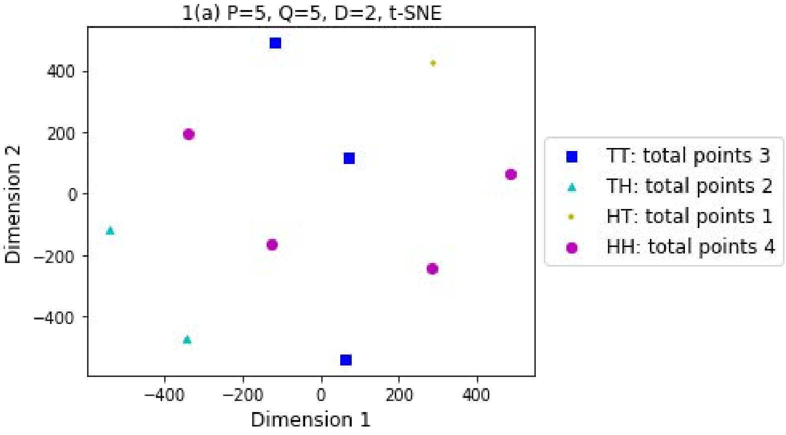
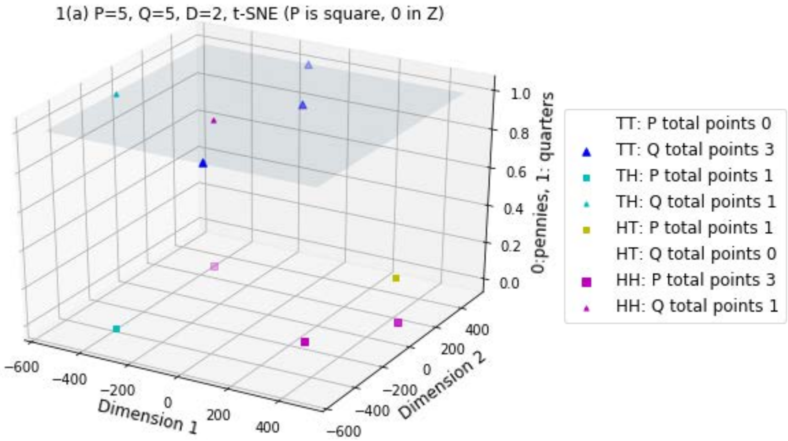


Figure: P and Q are seperated



(b) Simulate sequences of coin flips for $P = 10$ pennies and $Q = 10$ quarters. Visualize the data using t-SNE, clearly denoting the data points that are associated with common coin flip sequences.

Figure: P and Q are not seperated

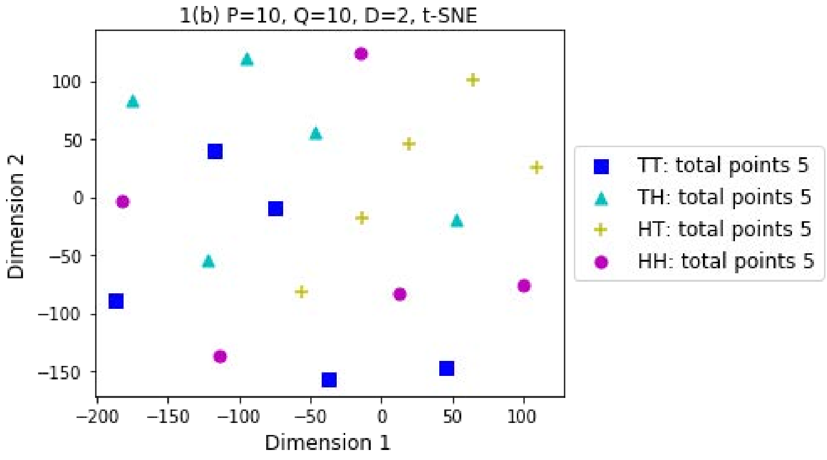
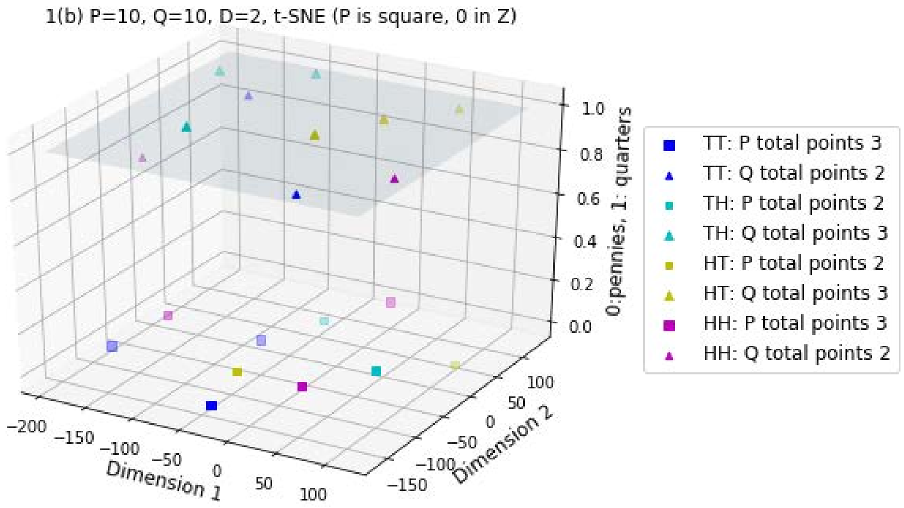
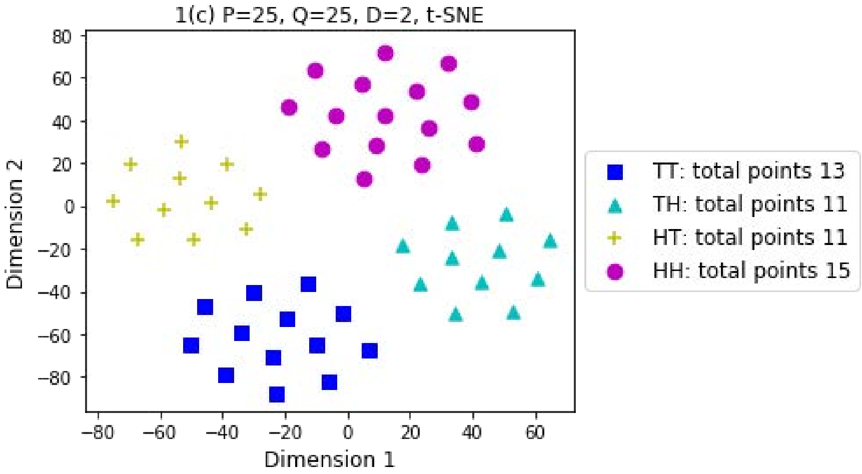


Figure: P and Q are seperated

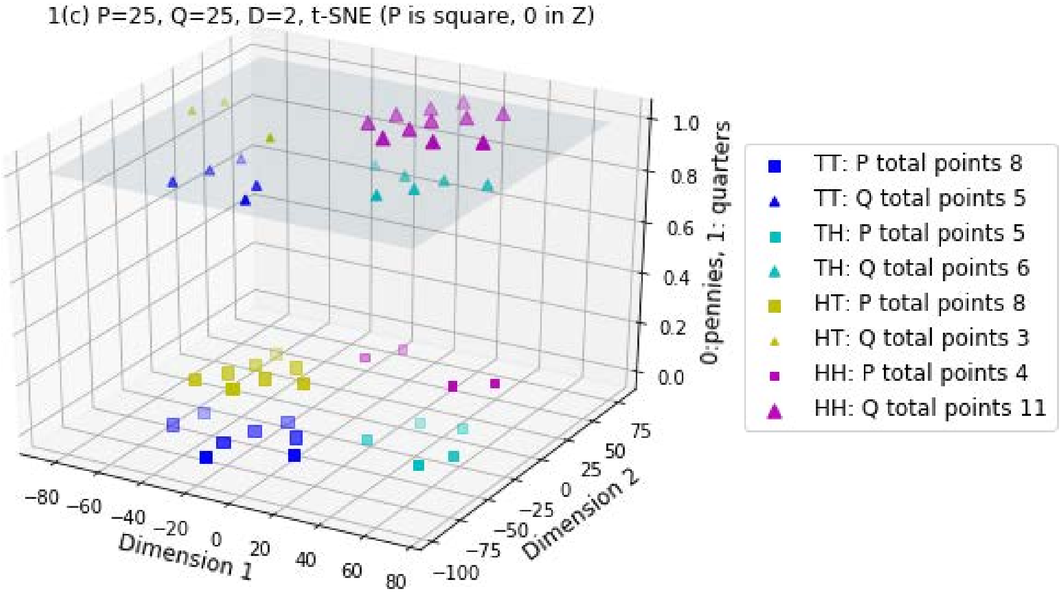


(c) Simulate sequences of coin flips for $P = 25$ pennies and $Q = 25$ quarters. Visualize the data using t-SNE, clearly denoting the data points that are associated with common coin flip sequences.

P and Q are not seperated

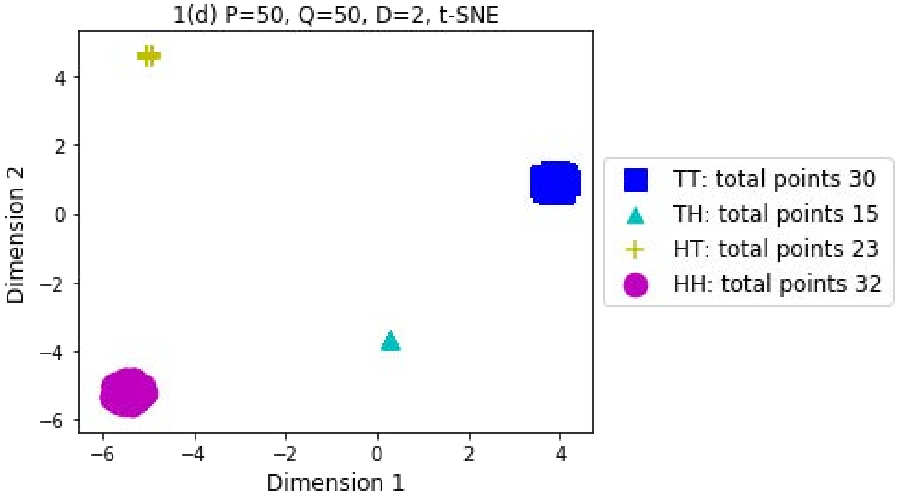


P and Q are seperated

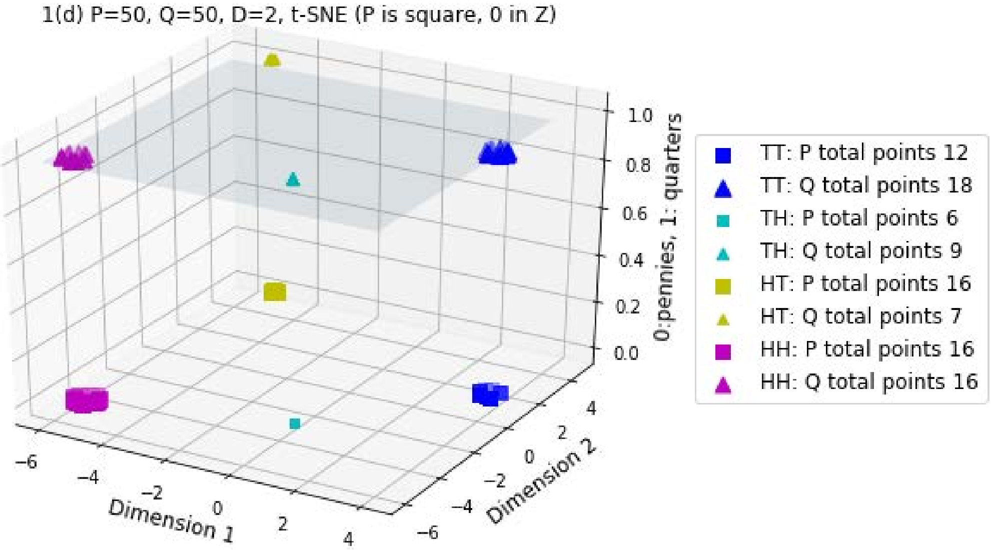


(d) Simulate sequences of coin flips for $P = 50$ pennies and $Q = 50$ quarters. Visualize the data using t-SNE, clearly denoting the data points that are associated with common coin flip sequences.

P and Q are not seperated



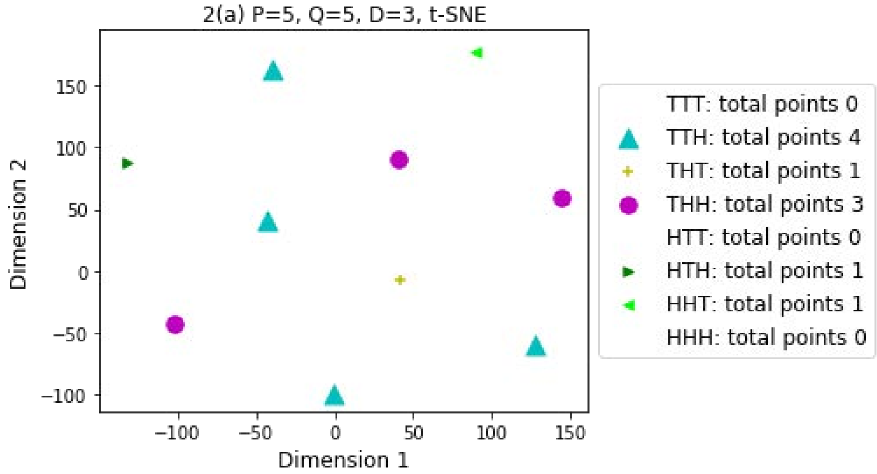
P and Q are seperated



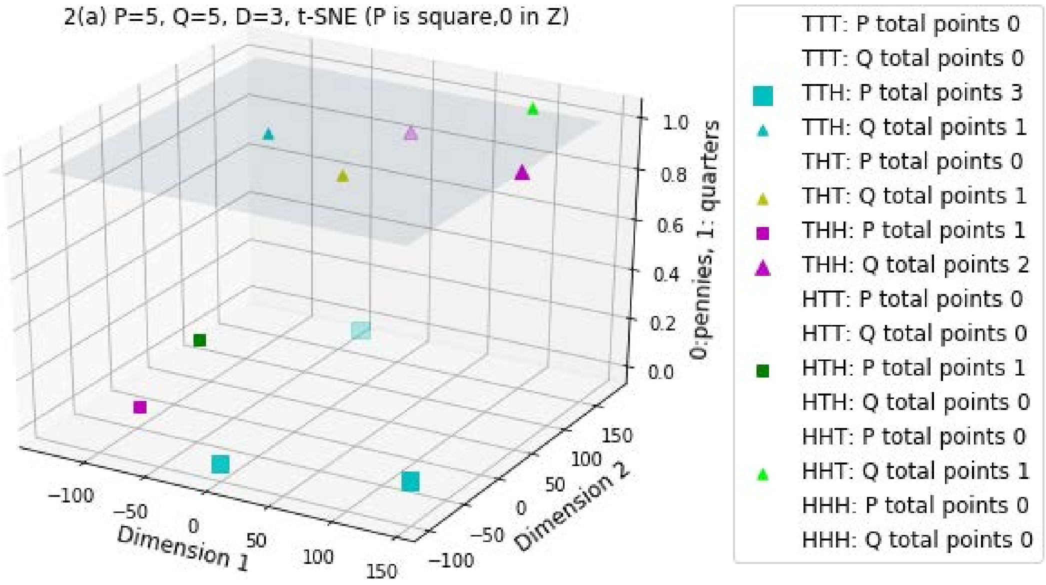
(20) 2. Now consider first the case where each coin is flipped three times ($D = 3$).

(a) Simulate sequences of coin flips for $P = 5$ pennies and $Q = 5$ quarters. Visualize the data using t-SNE, clearly denoting the data points that are associated with common coin flip sequences.

P and Q are not seperated

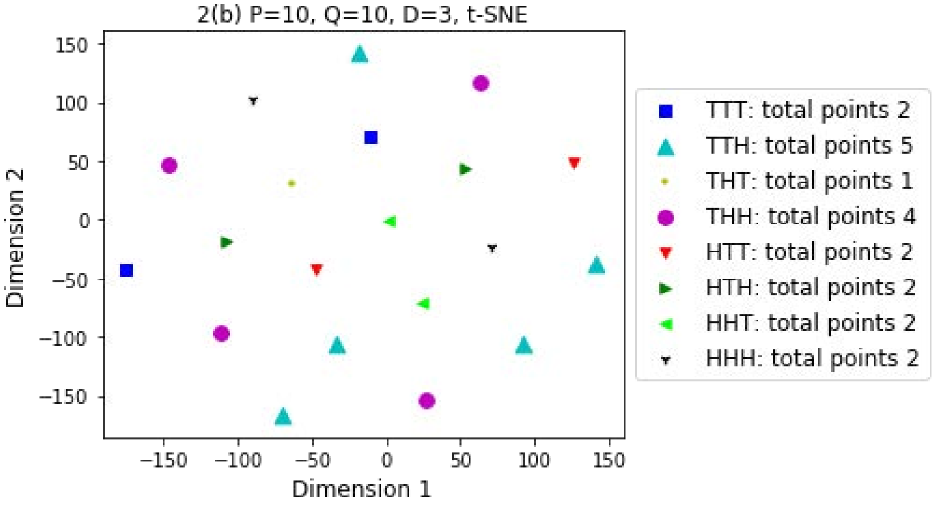


P and Q are seperated

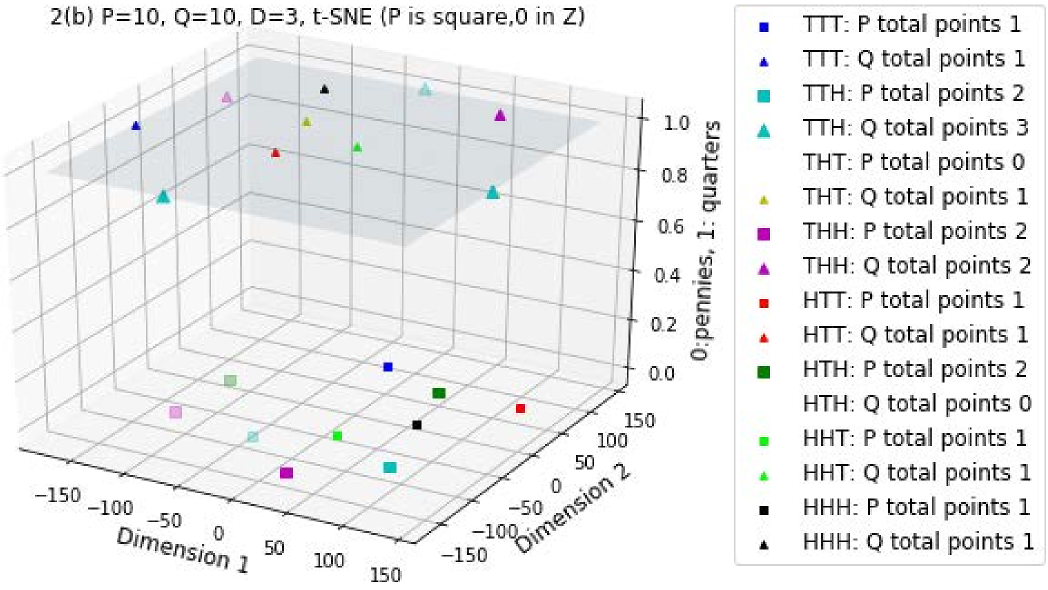


(b) Simulate sequences of coin flips for $P = 10$ pennies and $Q = 10$ quarters. Visualize the data using t-SNE, clearly denoting the data points that are associated with common coin flip sequences.

P and Q are not seperated

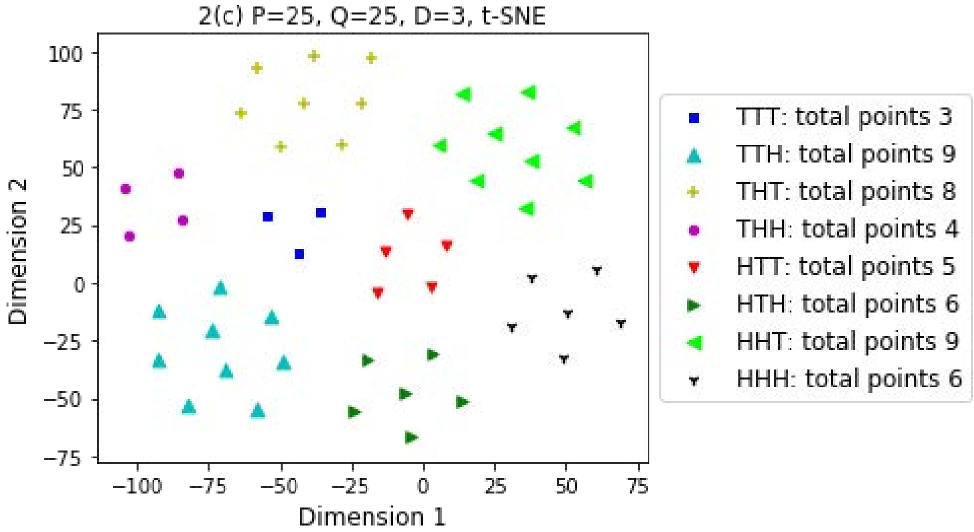


P and Q are seperated

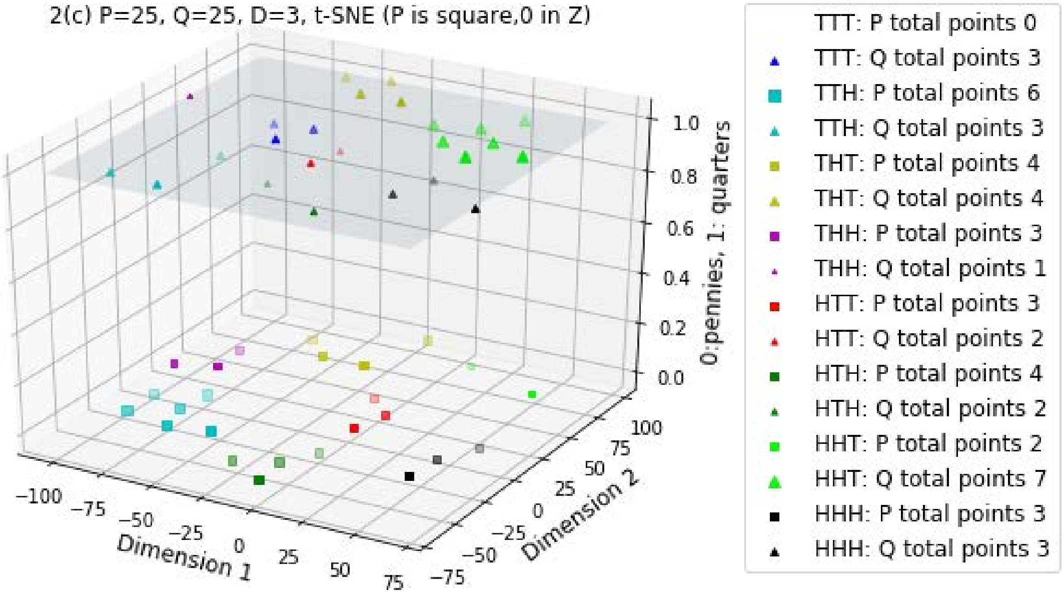


(c) Simulate sequences of coin flips for $P = 25$ pennies and $Q = 25$ quarters. Visualize the data using t-SNE, clearly denoting the data points that are associated with common coin flip sequences.

P and Q are not seperated

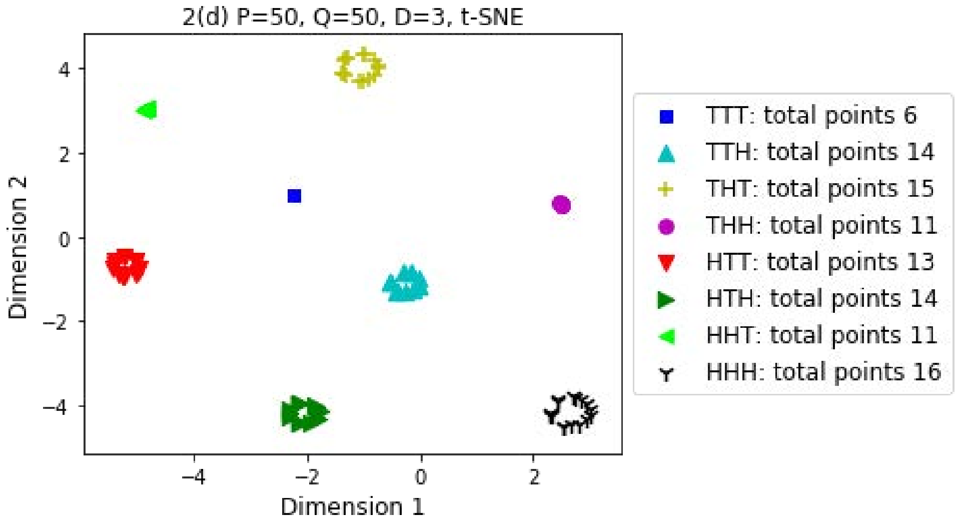


P and Q are seperated

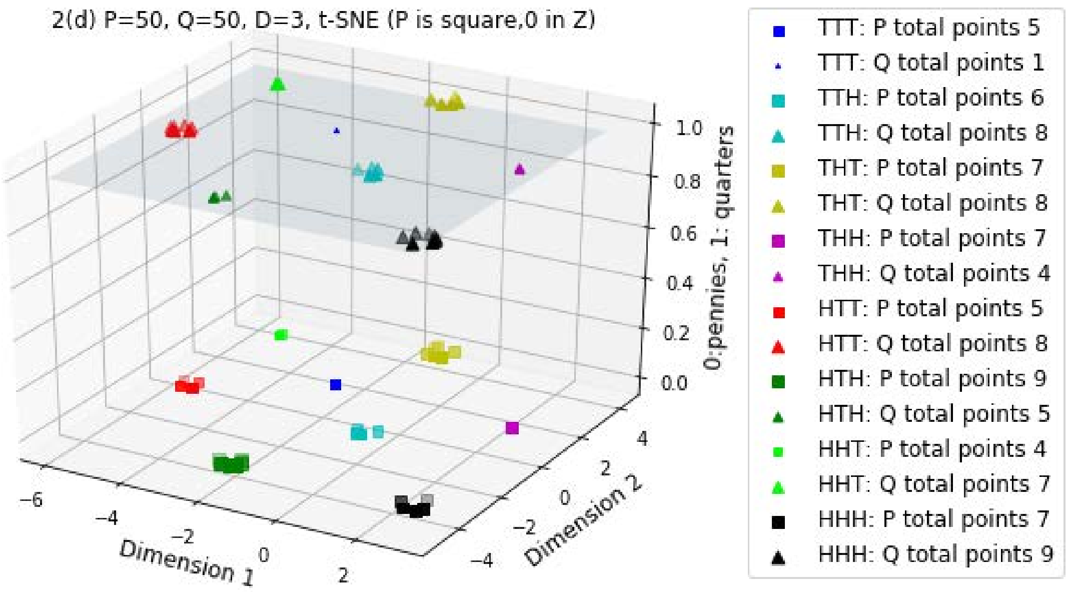


(d) Simulate sequences of coin flips for $P = 50$ pennies and $Q = 50$ quarters. Visualize the data using t-SNE, clearly denoting the data points that are associated with common coin flip sequences.

P and Q are not seperated



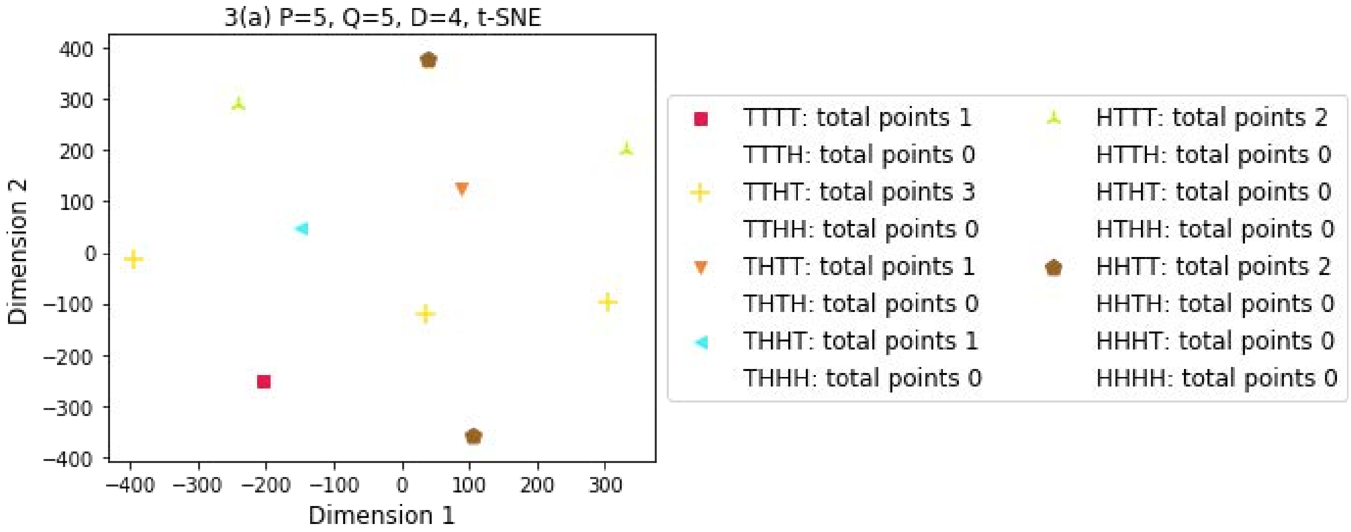
P and Q are seperated



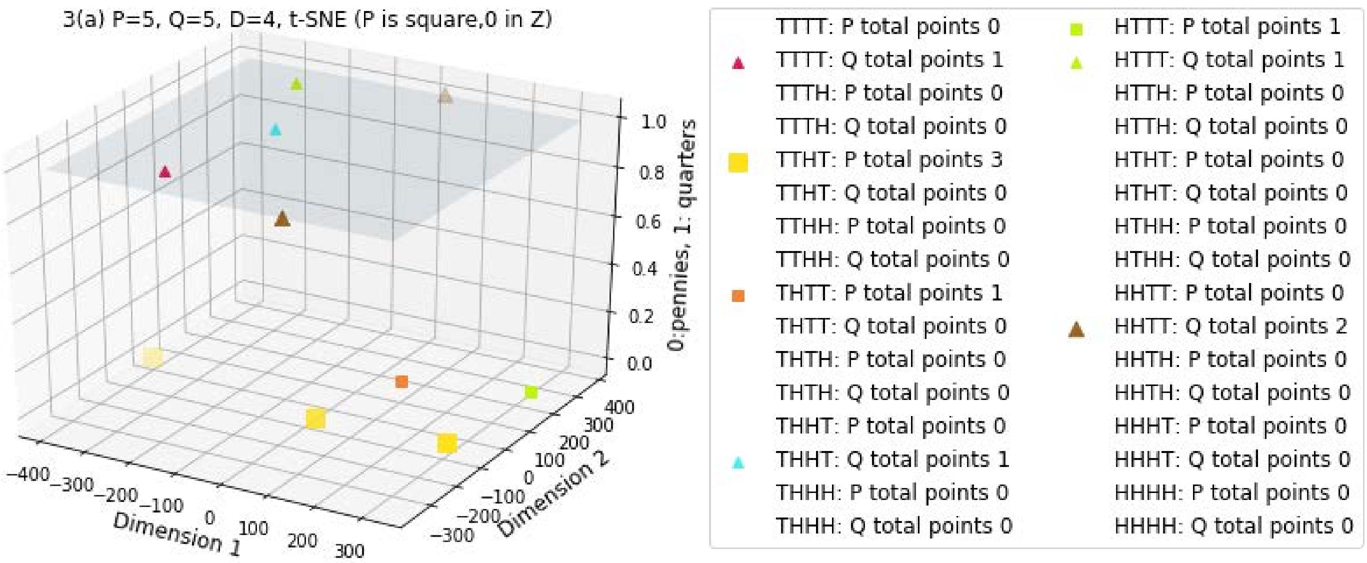
(20) 3. Now consider first the case where each coin is flipped four times ($D = 4$).

(a) Simulate sequences of coin flips for $P = 5$ pennies and $Q = 5$ quarters. Visualize the data using t-SNE, clearly denoting the data points that are associated with common coin flip sequences.

P and Q are not seperated

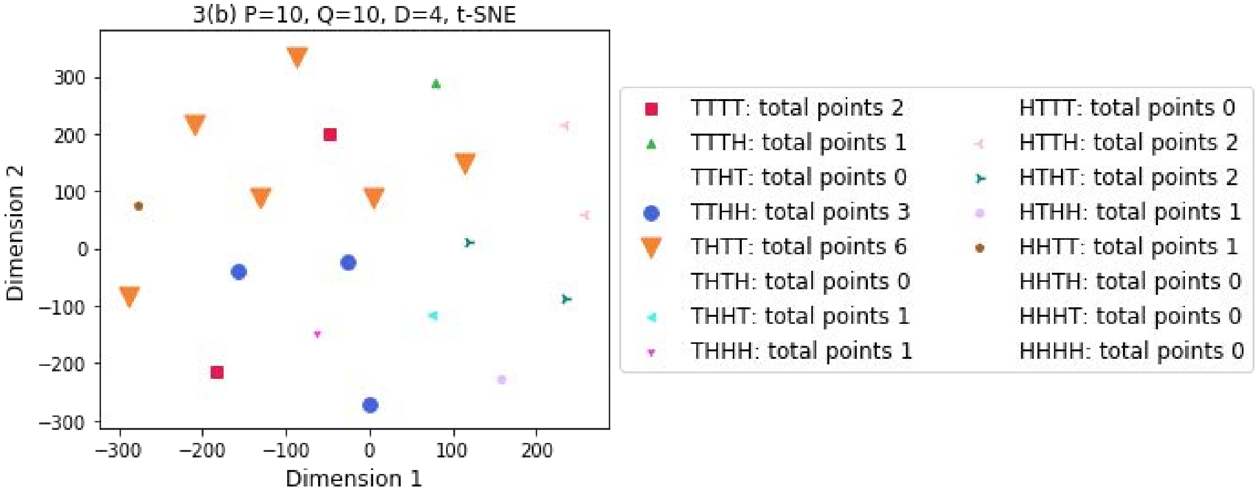


P and Q are seperated

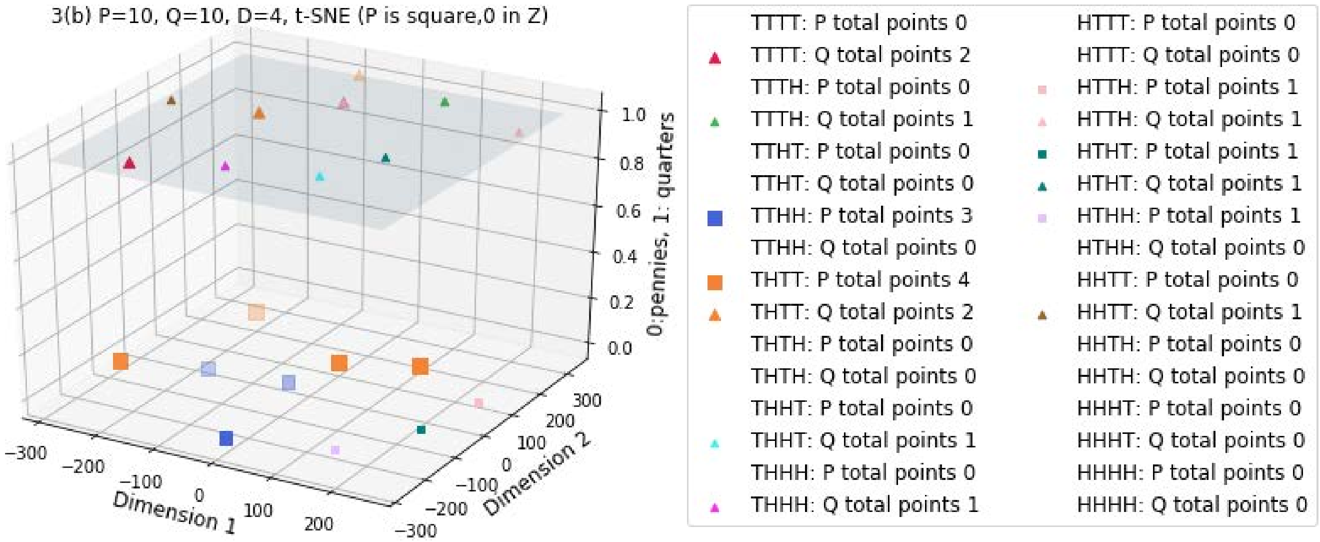


(b) Simulate sequences of coin flips for $P = 10$ pennies and $Q = 10$ quarters. Visualize the data using t-SNE, clearly denoting the data points that are associated with common coin flip sequences.

P and Q are not seperated

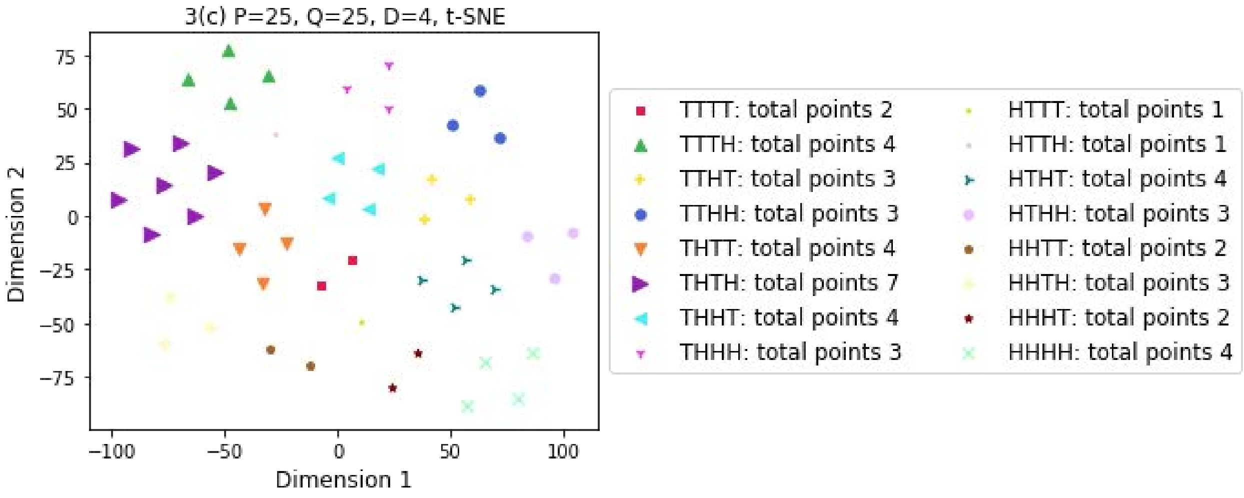


P and Q are seperated

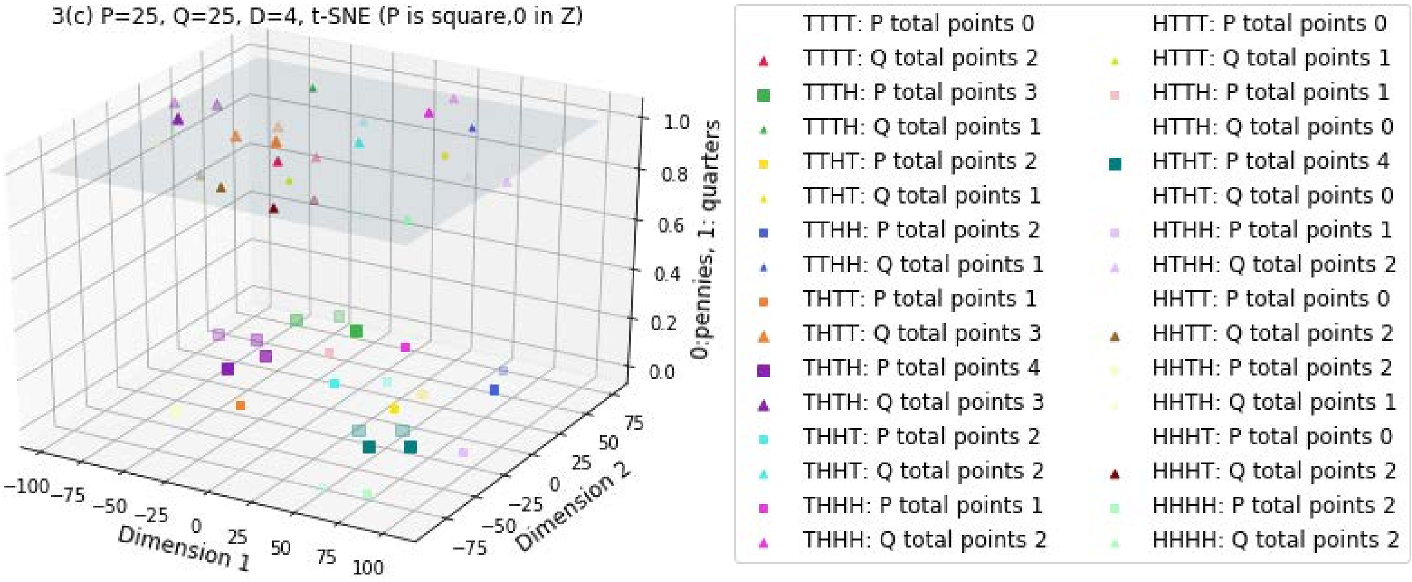


(c) Simulate sequences of coin flips for $P = 25$ pennies and $Q = 25$ quarters. Visualize the data using t-SNE, clearly denoting the data points that are associated with common coin flip sequences.

P and Q are not seperated

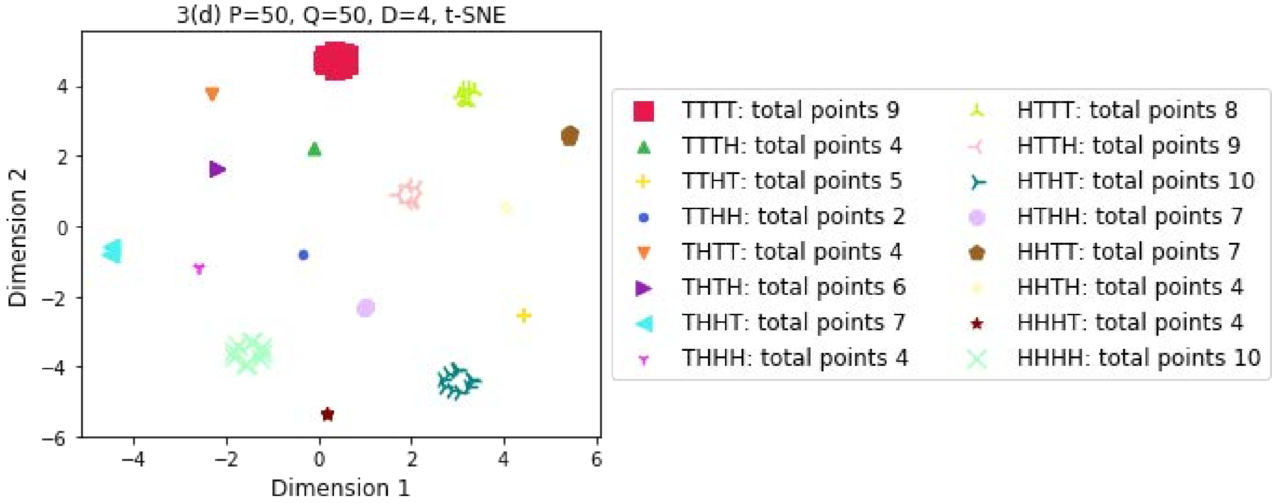


P and Q are seperated

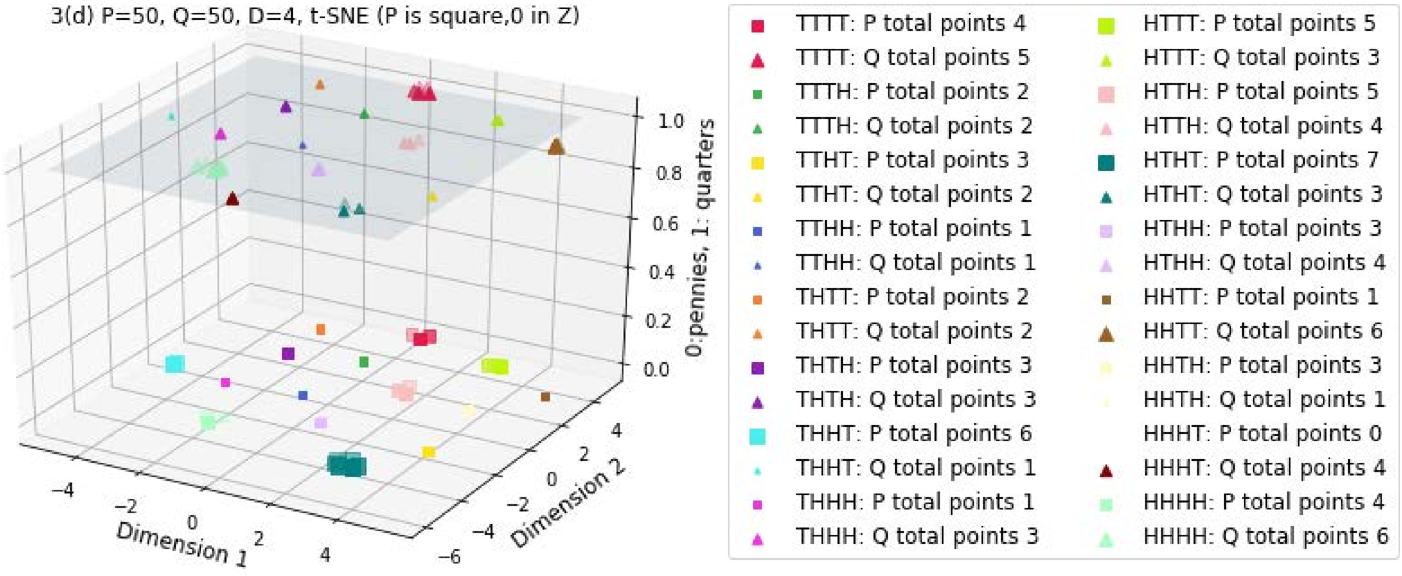


(d) Simulate sequences of coin flips for $P = 50$ pennies and $Q = 50$ quarters. Visualize the data using t-SNE, clearly denoting the data points that are associated with common coin flip sequences.

P and Q are not seperated



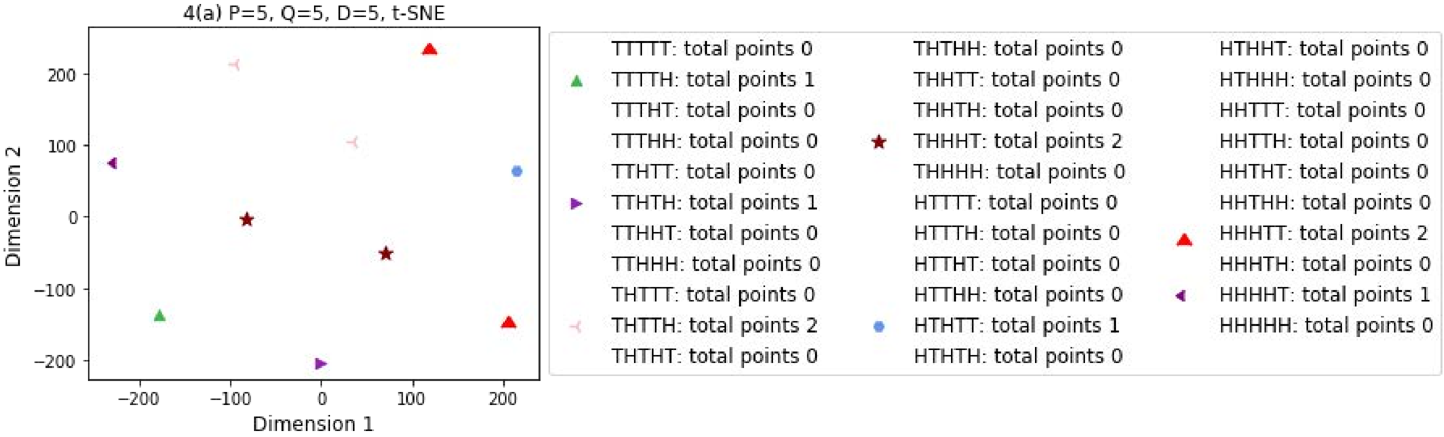
P and Q are seperated



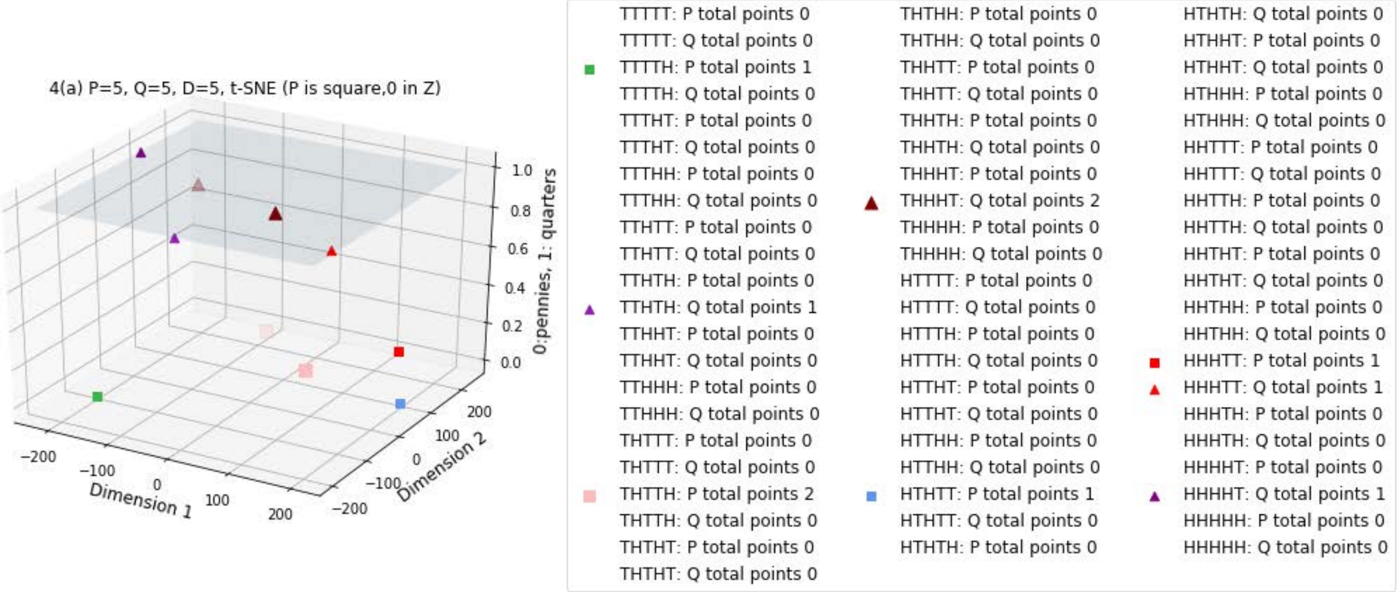
(20) 4. Now consider first the case where each coin is flipped four times ($D = 5$).

(a) Simulate sequences of coin flips for $P = 5$ pennies and $Q = 5$ quarters.[2] Visualize the data using t-SNE, clearly denoting the data points that are associated with common coin flip sequences.

P and Q are not seperated



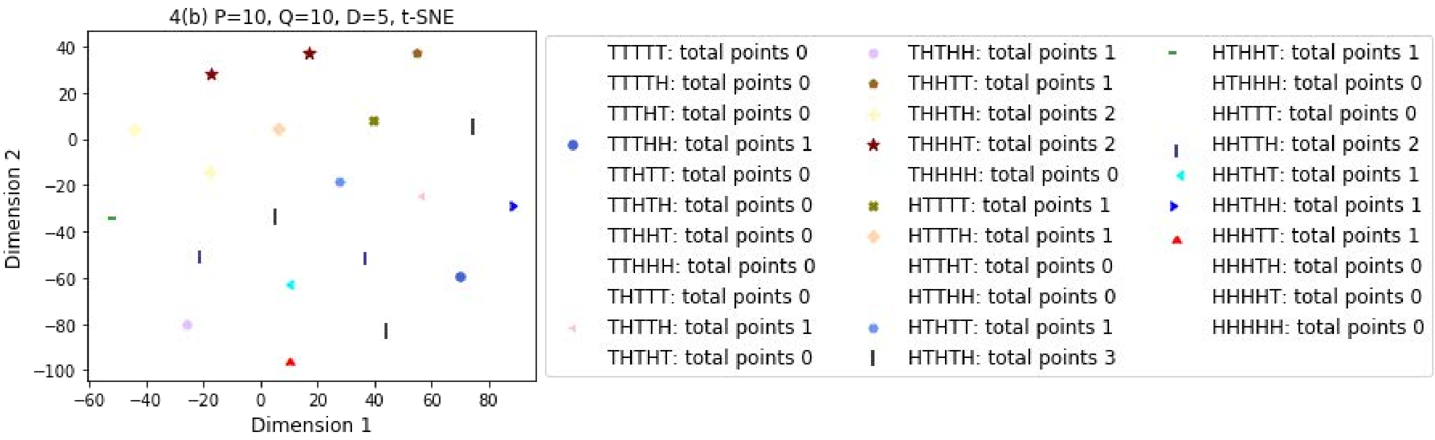
P and Q are seperated



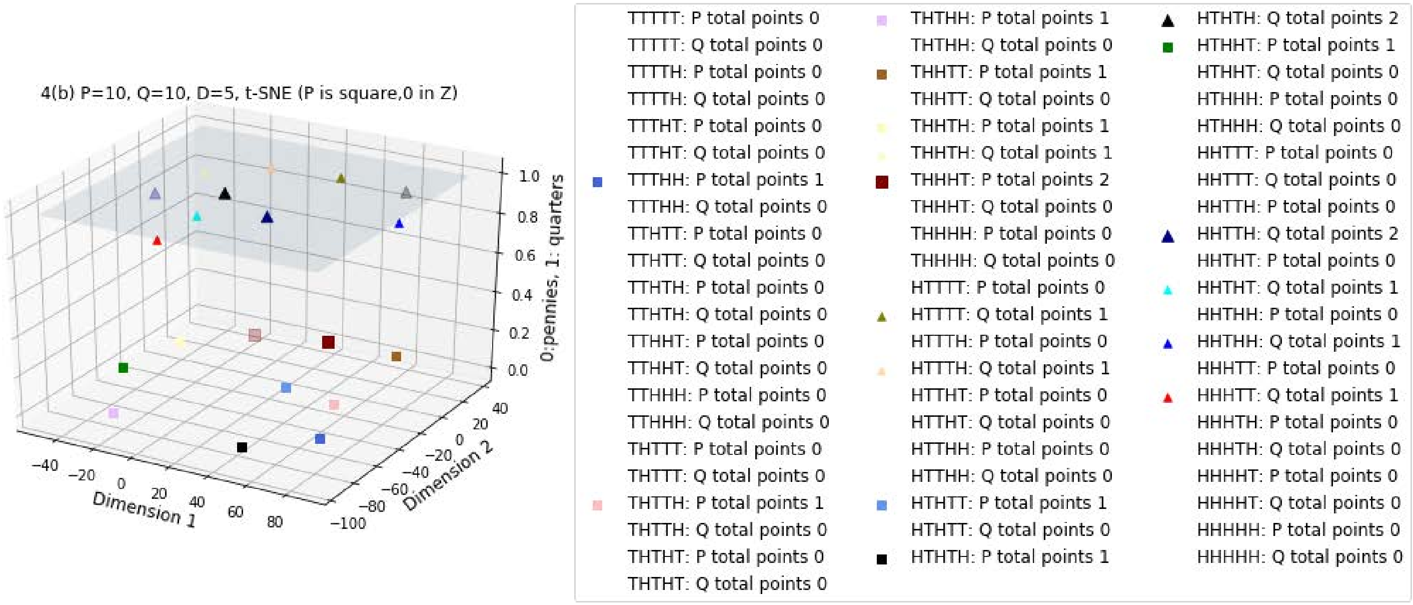
[2] This is the scenario which you explored via Monte Carlo simulation.

(b) Simulate sequences of coin flips for $P = 10$ pennies and $Q = 10$ quarters. Visualize the data using t-SNE, clearly denoting the data points that are associated with common coin flip sequences.

P and Q are not seperated

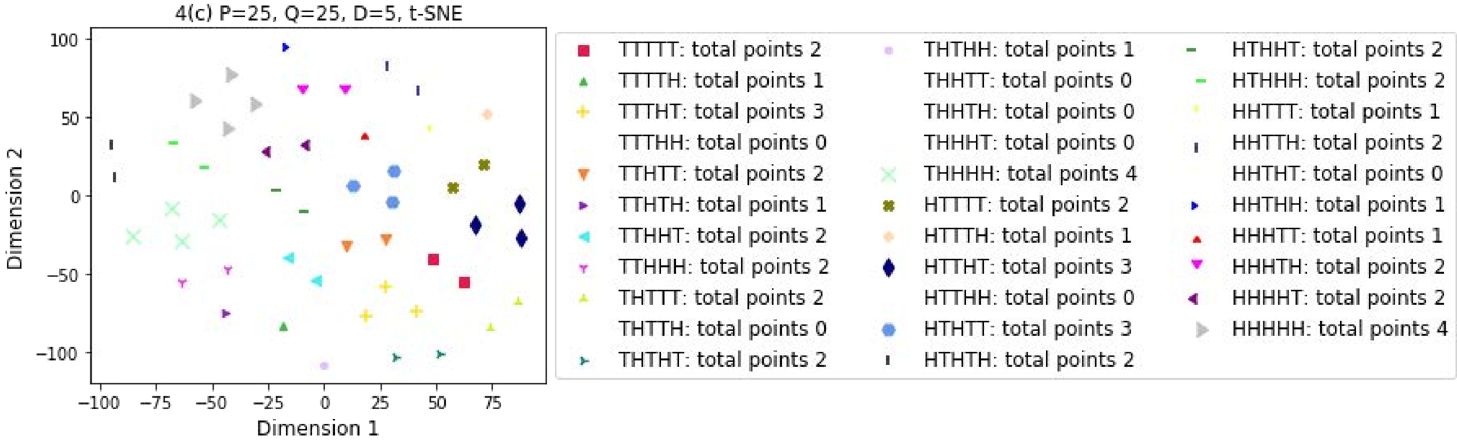


P and Q are seperated

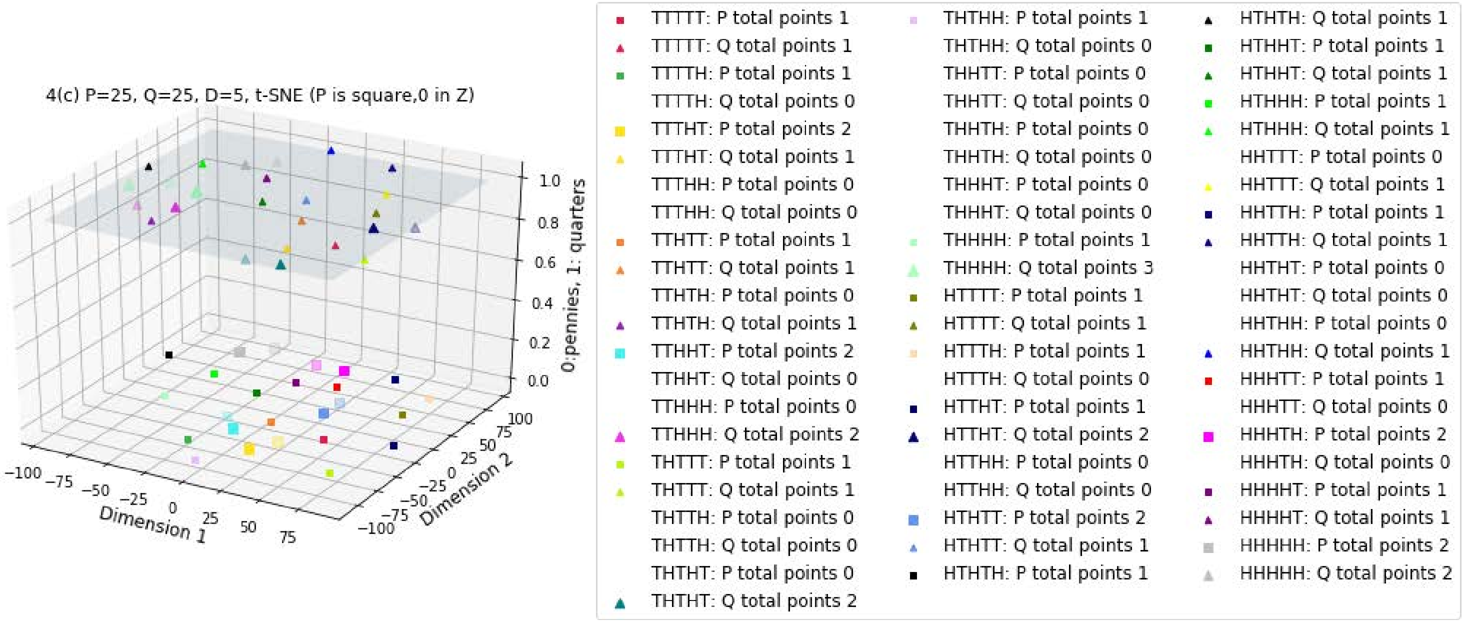


(c) Simulate sequences of coin flips for $P = 25$ pennies and $Q = 25$ quarters. Visualize the data using t-SNE, clearly denoting the data points that are associated with common coin flip sequences.

P and Q are not seperated

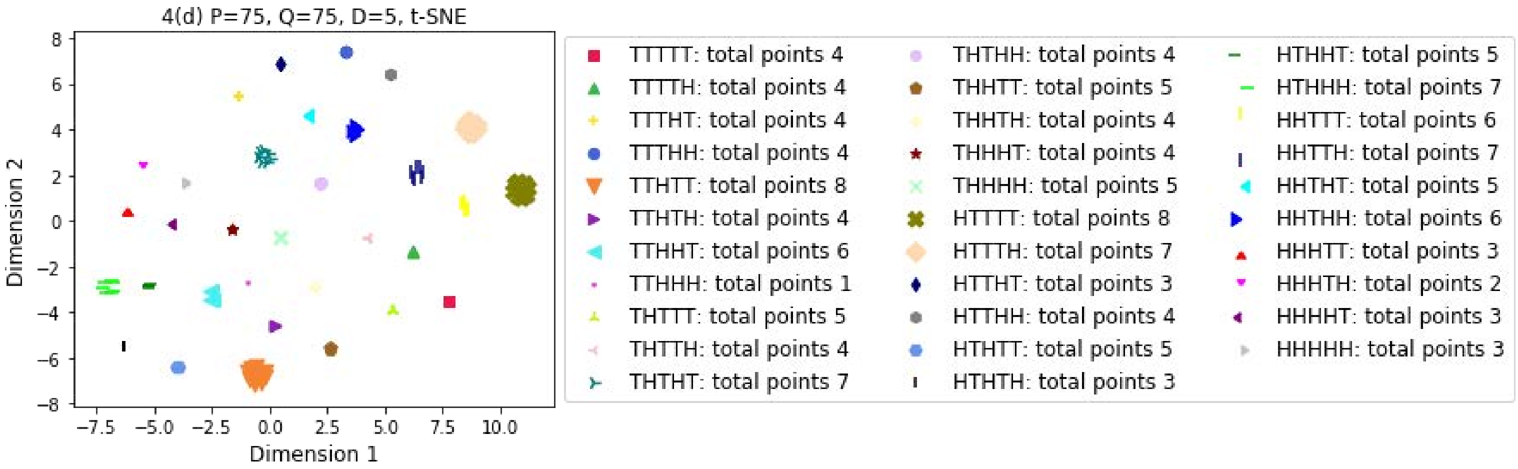


P and Q are seperated

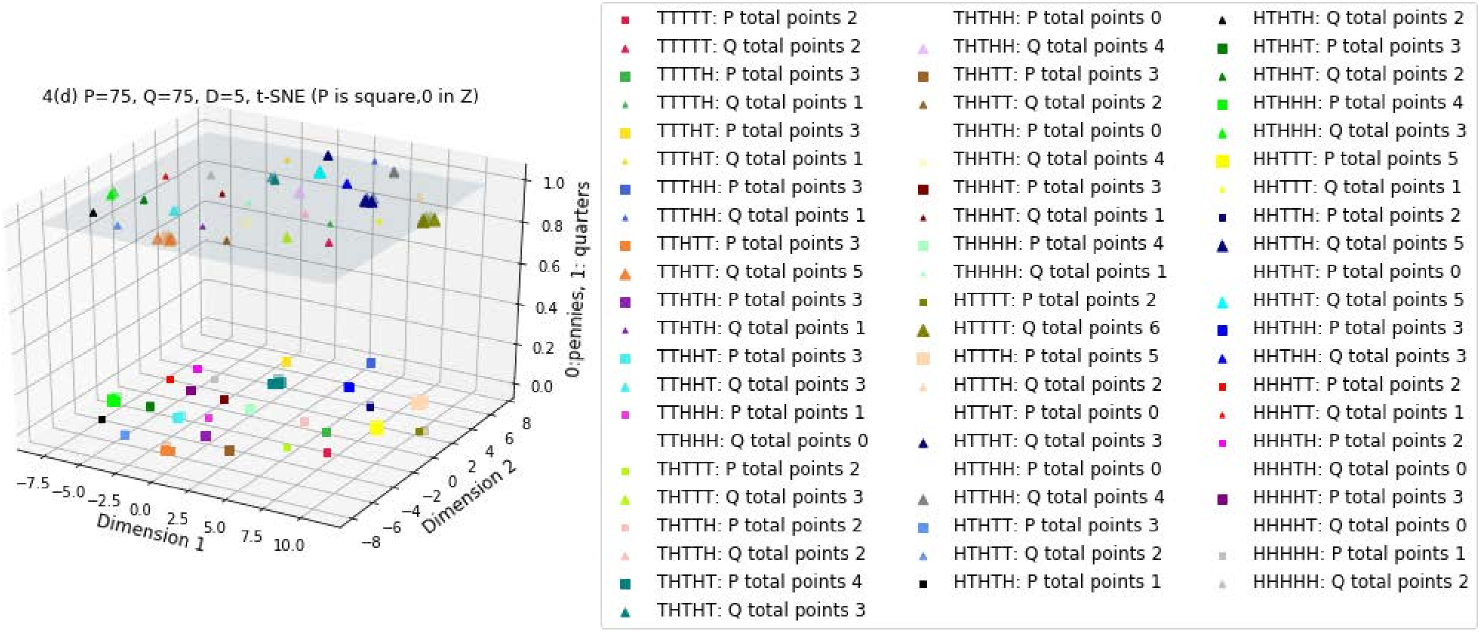


(d) Simulate sequences of coin flips for $P = 75$ pennies and $Q = 75$ quarters. Visualize the data using t-SNE, clearly denoting the data points that are associated with common coin flip sequences.

P and Q are not seperated



P and Q are seperated



(30) 5. (a) What trend do you see in the clusters produced by t-SNE as the number of observations[3] increases?

Solution:
Under the same number of features, as the number of observations increases, the data points that are associated with common coin flip sequences are more likely to be clustered. Different classes are more likely to be separable and clustered.

(b) What trend do you see in the clusters produced by t-SNE as the number of features[4] increases?

Solution:
Under the same number of observations, as the number of features increases, the data points that are associated with common coin flip sequences are less likely to be clustered. It becomes difficult to separate and cluster different classes.

(c) How might changes in the number of observations and features impact the conclusions drawn from the visualizations produced by t-SNE, particularly in the absence of ground truth to help guide interpreting the visualization?[5]

Solution:
As the number of observations increases and the number of features decreases, the data points that are associated with common coin flip sequences are more likely to be clustered. It becomes easier to separate and cluster different classes, vice versa.
This is because when the number of observations increases, the probability of seeing identical(or substantially similar, in the case of continuous variables) data under both classes increases. And when the number of features increases, the probability of seeing identical (or substantially similar, in the case of continuous variables) data under both classes decreases.
In the absence of ground truth, it will make it even harder to distinguish different classes when the number of observations decreases and the number of features increases.

[3] In the distinguishing coins problem, the number of coins (P and Q) is the number of observations.
[4] In the distinguishing coins problem, the number of flips of each coin (D) is the number of features.
[5] Feel free to run Monte Carlo simulations before answering this question, if you think that would be helpful.