MTH511A Graded Assignment Report

Akshay Bhatia

1 November 2018

Note:
This document contains
1. Observations for numerical experiments for all the questions.
2. Mathematical details for questions 4 and 6.
In detail proofs and theoretical explanations for Question 1(b) and Question 5 are attached in the supplementary document.
All the results and figures will be generated on running the code.

1 Question 1(a)

Instead of a square, if we consider the smallest equilateral triangle containing the circle and generate observations from a uniform distribution on the triangle, it will **NOT** improve upon the estimate of pi obtained from the square. The sample output of my code testifies this statement.

Estimated value of pi from
1. Square = 3.142400
2. Triangle = 3.127564

Percentage Error in value of pi from
1. Square = 0.025699
2. Triangle = 0.446541

Fraction of points rejected
1. Square = 0.214400
2. Triangle = 0.398100

2 Question 1(b)

To estimate the value of e, the following methods were used. Please consider that the methods which explicitly use log or exponential function were avoided (Method 1. being exception)

Monte Carlo based methods:-
1. Inverse transform method
2(a). Extreme value distribution(EVD)
2(b). Variance reduction in EVD method using antithetic variables.
3. Bootstrap method
4. Order statistic and extreme value distribution.
5. Definiton of pdf of standard normal distribution.

Numerical approximations:-
Three series which converge to e were used.

Sample output from the code is

Estimated value of e from Monte Carlo Techniques
1. Method 1: 2.721318
2. Method 2a: 2.717790
3. Method 2b: 2.718355
4. Method 3: 2.725899
5. Method 4: 2.714898
6. Method 5: 2.742827

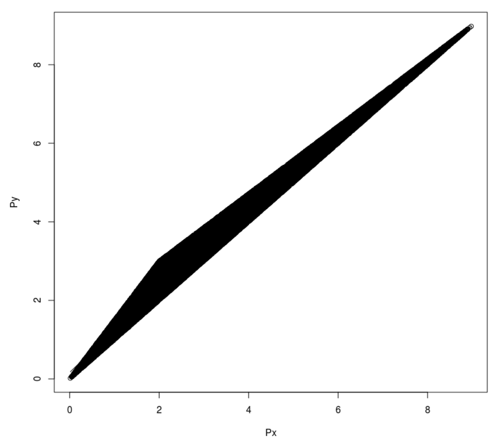
Estimated value of e from Numerical Approximations
7. Method 1: 2.718295
8. Method 2: 2.718266
9. Method 3: 2.718284

3 Question 2

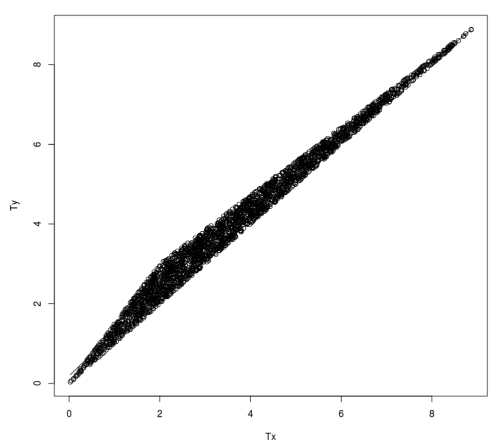
WLOG, we can simulate uniform samples from a triangle with one vertex at origin.
The demo triangle from which the points were sampled had coordinates

(0,0), (2,3), (9,9).
Total number of uniform points used was set to n = 50000

3.1 Triangle - Direct geometrical method



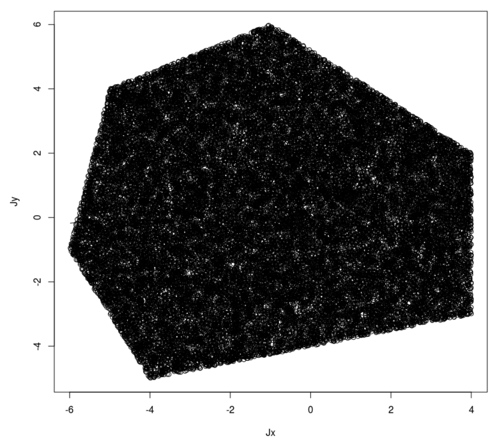
3.2 Triangle - Accept Reject method



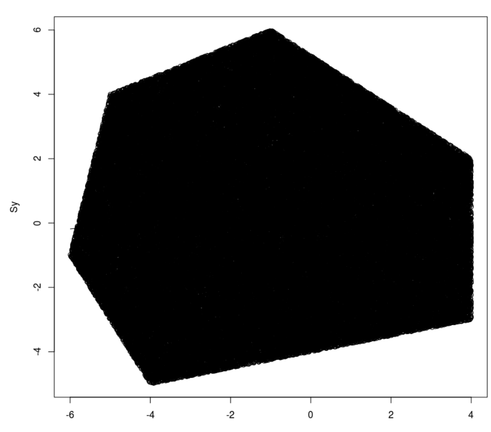
To simulate from a polygon, we first divide into a section of triangles by joining each vertex to origin. Then, we can apply alias method to decide the triangle from which to simulate, weighted by its area. The demo convex polygon(six-sided) from which the points

were sampled had coordinates
(-1,6),(4,2),(4,-3),(-4,-5),(-6,-1),(-5,4)
Total number of uniform points used was set to n = 50000

3.3 Polygon - Accept Reject method



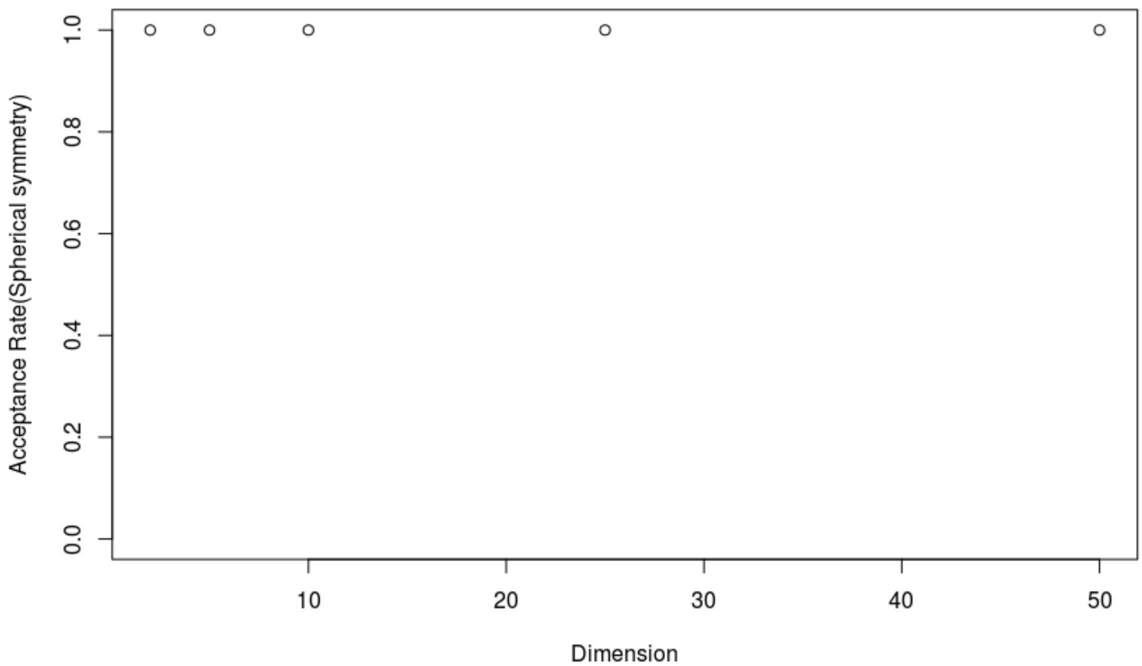
3.4 Polygon - Alias Method + Geometrical Method



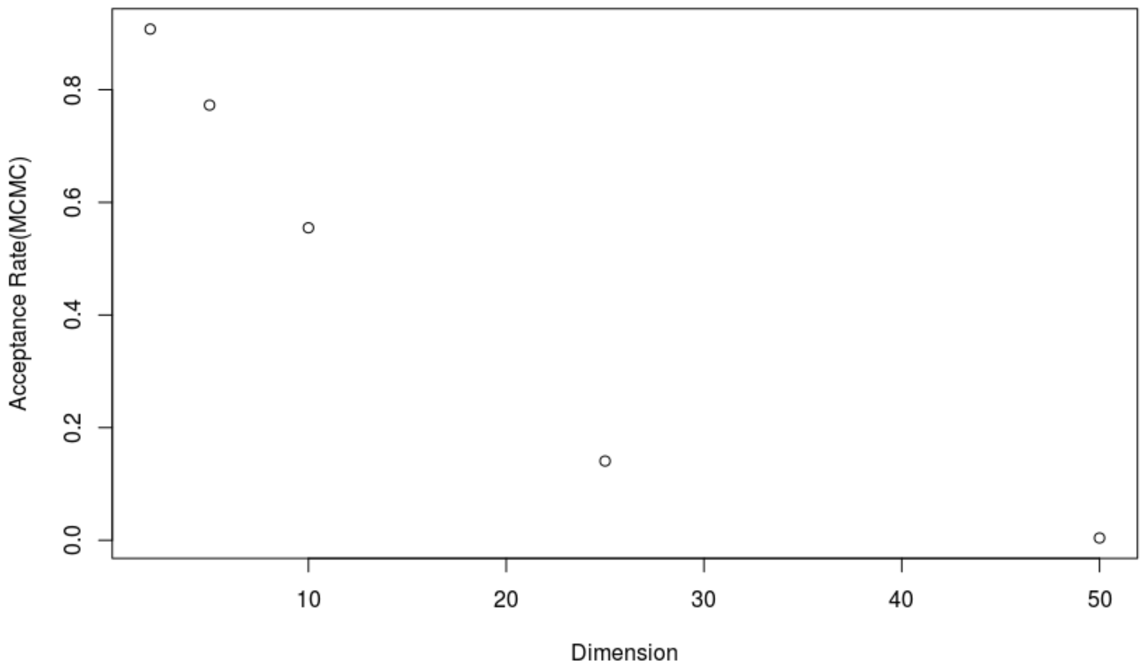
4 Question 3

The plots of acceptance ratio versus dimension is shown for the three methods.

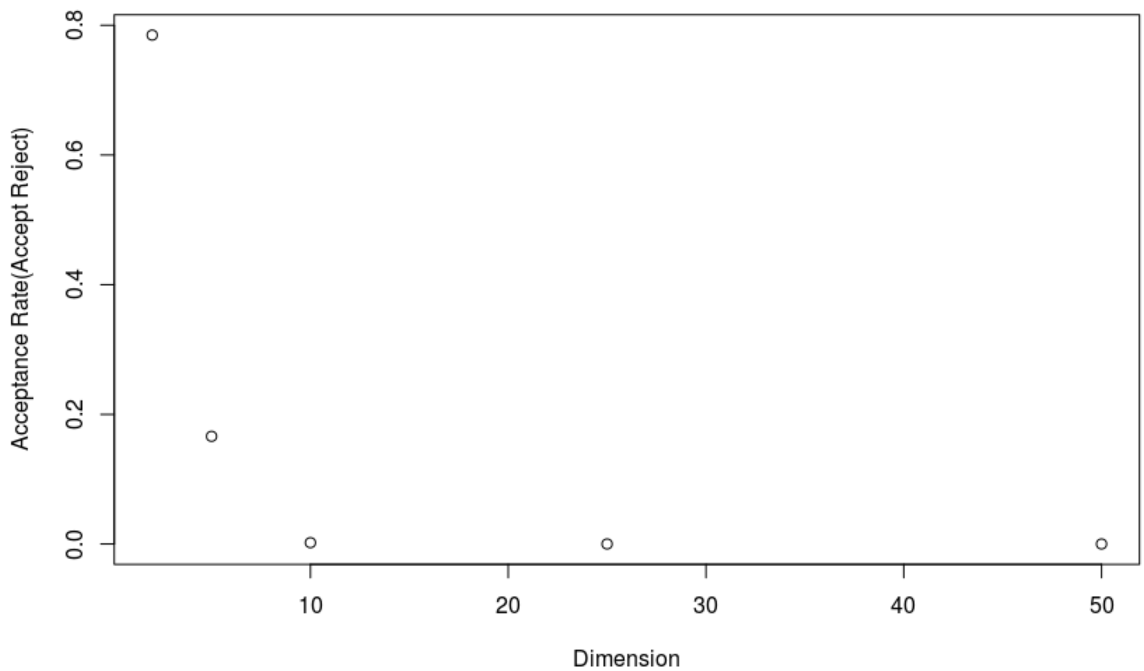
4.1 Spherical Symmetry



4.2 MCMC Method



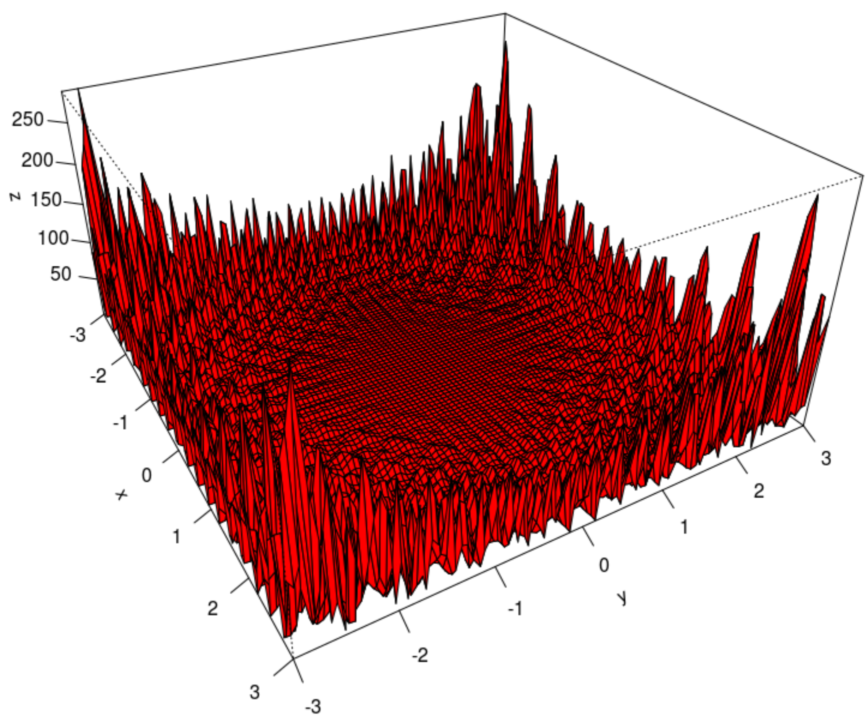
4.3 Accept Reject



Acceptance for Accept Reject method drops close to zero for dim close to 10.

5 Question 4

The function plot to be optimised can be visualised as:



The objective function is given by :

$$(xsin(20y)+ysin(20x))^2cosh(sin(10x)x)+(xcos(10y)ysin(10x))^2cosh(cos(20y)y)$$

Consider the constituent terms of the function - Both of the are independently postive, since square function as well as *cosh* function are positive $\forall x, y \in R$.
Therefore, minimum value which the function can attain is zero at (0,0).

Following algorithms were used to find the global minima:

Deterministic:
1. Simple grid search
2. Gradient Descent method
3. Nelder Mead Simplex search
4. BFGS method

Monte Carlo Algorithms:
1. Pure random optimisation
2. Genetic Algorithm
3. Simulated annealing.

All the **deterministic algorithms** perform terribly on the given function.
Only reasonable estimates of the global minima are obtained once the initialisation has been done very close to the known global minima(0,0).

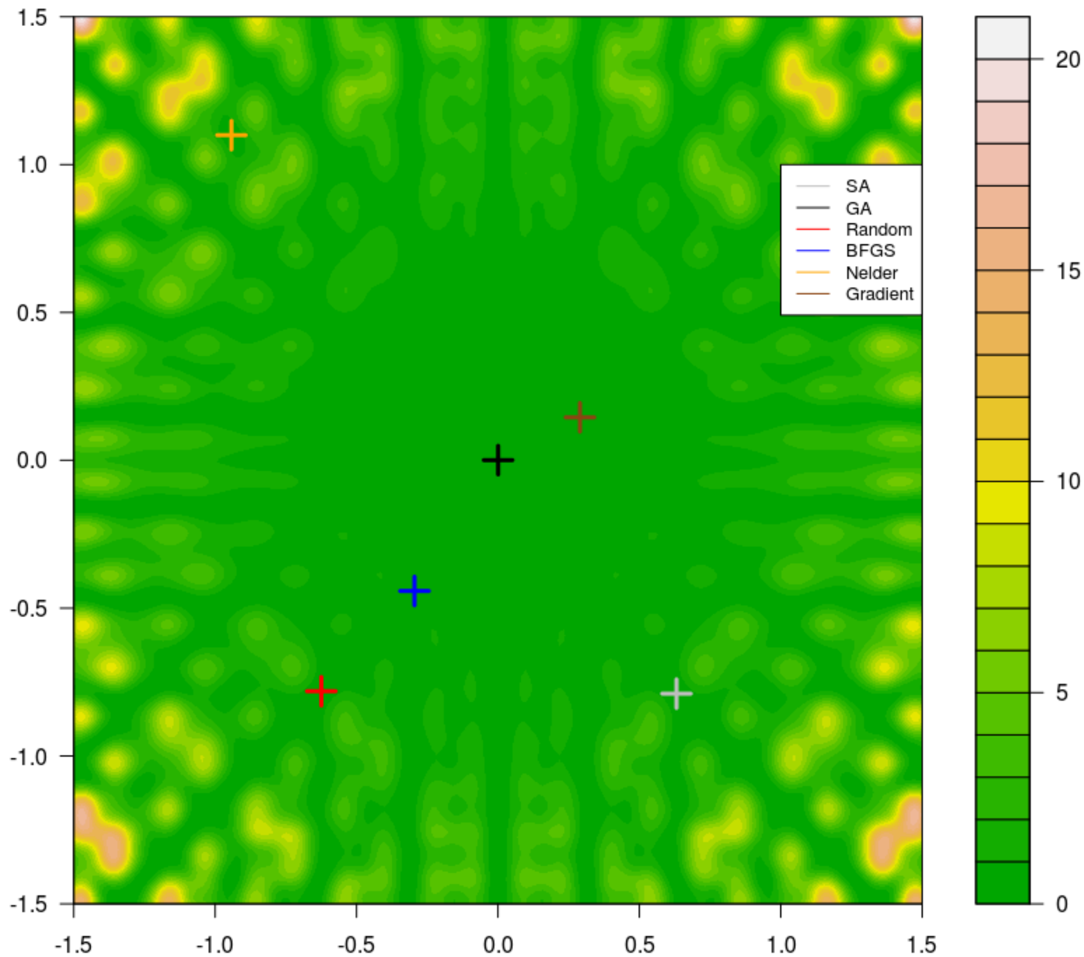
On the performance of Monte Carlo optimisation:-

Pure random optimisation performs really well for almost all starting values of x, y but gets stuck in a local minima which is close to global minima.

Simulated annealing performs well but relatively poorly to random optimisation and needs initialisation which is close to global minima.

Genetic algorithm is by far the most robust algorithm out of all the algorithms used, and converges to global minima for almost all initial values of global minima.

The global minima returned by above algorithms is plotted in contour map shown below.



The pros and cons of each algorithm are stated in the supplementary document.

6 Question 5

For performing simulation study, following value of initial parameters were used.
n1 = 200,
n2 = 40,
n3 = 10,
N = 375
(according to notation given in question)
Intial guess for allele frequencies
pa = 0.7
pb = 0.1
po = 0.2

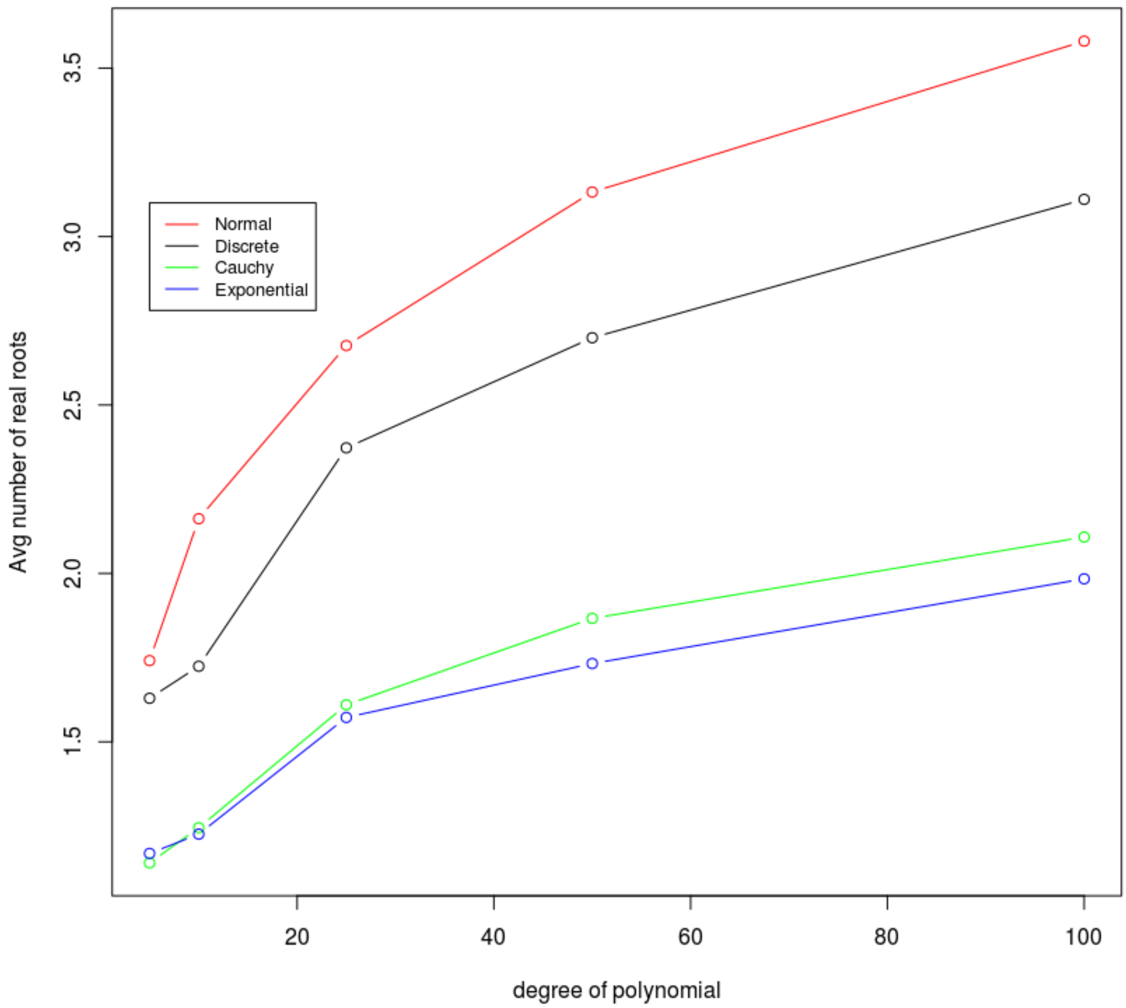
Allele frequencies as estimated by EM algorithm and Fisher's Information algorithm:

EM Algorithm:
pa = 0.33952979
pb = 0.06963494
po = 0.59083527

Fisher's Scoring Algorithm:
pa = 0.33951682
pb = 0.06963458
po = 0.59084860

Also EM converged in 7 iterations while Fisher converged in 5 iterations.
Which is also illustrated in supplemetary document.

7 Question 6



$$p_n(t) = X_0 + X_1 t + ... + X_n t^n$$

Let the number of real roots of $p_n(t)$ be denoted by A_n. The above plot shows the variation of $E(A_n)$ versus n. It can be observed that approximate log behaviour is shown by $E(A_n)$ for all the distributions.

To find the average number of real roots of polynomial equations, simply run the code for Question 6.

	Discrete	**Normal**	**Cauchy**	**Exponential**
Degree = 5	1.625	1.777	1.137	1.153
Degree = 10	1.727	2.159	1.241	1.234
Degree = 25	2.379	2.681	1.601	1.569
Degree = 50	2.710	1.135	1.871	1.766
Degree = 100	3.164	3.559	1.994	1.958

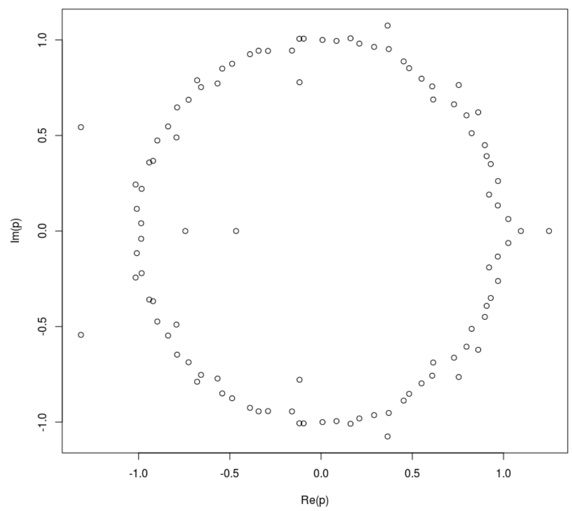
For the polynomial equations with coefficients from
1. $\{-1, 1\}$ with equal probability
2. $N(0, 1)$
It was observed that most of the real roots of the equations cluster around the points 1 and —1.

No such pattern was observed for iid coefficients from $C(0, 1)$ and $exp(1)$ distributions, probably implying that asymmetrical and heavy tailed distributions do not enjoy this property.

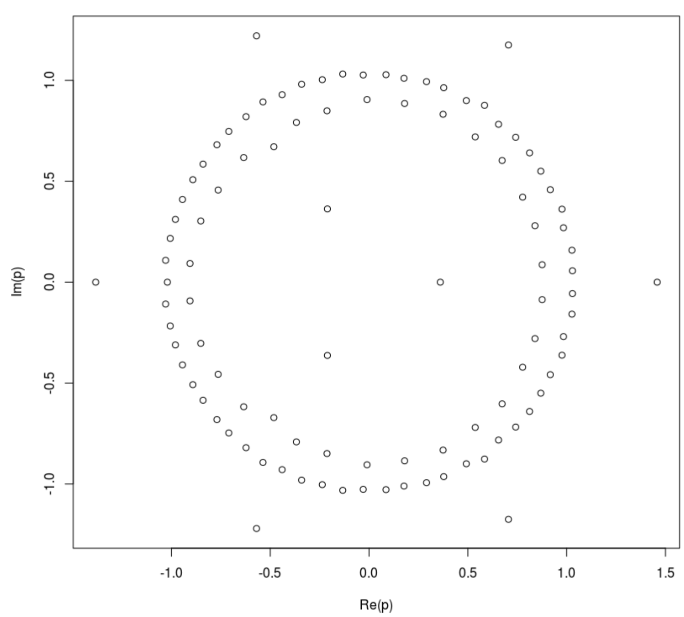
Also the the complex roots of $p_n(t)$ with i.i.d. coefficients $X_0, ..., X_n$ concentrate near the unit circle as $n \to \infty$.
Whereas, if the coefficients are heavy tailed then the roots concentrate on the union of two centered circles.

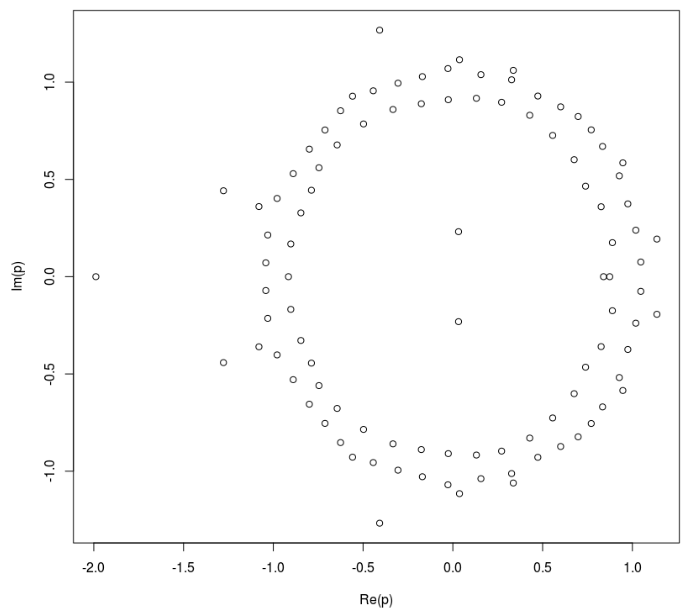
Below shown are the plots of complex roots of random polynomial.

For iid coefficients from uniform discrete, $N(0, 1)$ and $exp(1)$ polynomials, common pattern is followed by the roots as they concentrate near the unit circle:



For coefficients from iid $C(0,1)$, following pattern is formed by the roots





Consider a random polynomial $G_n(z) = \xi_n z^n + ... + \xi_1 z + \xi_0$ with i.i.d. cauchy coefficients.
The radii of the circles are $|\xi_0/\xi_\tau|^1/\tau$ and $|\xi_\tau/\xi_n|^1/(n-\tau)$, where ξ_τ denotes the coefficient with the maximum modulus.

Owing to the above result, we can find the expected number of roots which lie in a general subset of a complex plane.