Carnegie Mellon University
Department of Computer Science
15-826 Multimedia Databases and Data Mining
C. Faloutsos, Spring 2017

Homework 3 (by Yuang Liu) - Solutions

Due: **hard copy, in class, at 3:00pm, on 3/8/2017**

Due: **tarball, on Blackboard, at 3:00pm, on 3/8/2017**

VERY IMPORTANT:

- For each question, we expect *both* a **hard copy**, and a **tar file** with your code.
- Deposit **hard copy** of your answers, in class.
  1. **Separate** your answers, on different page(s) for each question
  2. **Type** the full info on **each** page: your **name**, **Andrew ID**, **course#**, **Homework#**, **Question#** on each of the pages.

Reminders:

- *Plagiarism*: Homework is to be completed *individually*.
- *Typeset* your answers. Illegible handwriting may get zero points.
- *Late homeworks*: please email it
  - to all TAs and graders
  - with the subject line exactly `15-826 Homework Submission (HW 3)`
  - and the count of slip-days you are using.

For your information:

- Graded out of **100** points; **4** questions total
- Rough time estimate: *16-24 hours (≈ 4-6 hours per question)*

Revision : 2017/03/23 01:26

Question	Points	Score
Density Paradox	15	
Correlation Integral	30	
Substring Editing Distance	25	
Percolation - phase transitions	30	
Total:	100	

Code packaging info: (identical to Homework 2, except for minor customizations: 'hw3' instead of 'hw2', etc)

Submit your code to blackboard, in a single `tar` file, called *`andrewId`*`-hw3.tar.gz` (where *andrewId* is your andrew id.) For your convenience, we provide a *tar-file package*, at `http://www.cs.cmu.edu/~christos/courses/826.S17/HOMEWORKS/HW3/hw3.tar.gz`. We will refer to it as the *tar-file package* from now on. It has 3 directories `/Q2, /Q3, /Q4`.

- `tar xvf hw3.tar.gz`; `cd Q2; make` # *to work on correlation integral*
- Replace the placeholder code with your solutions, `tar` everything into *`andrewId`*`-hw3.tar.gz` and submit to blackboard. We expect to do `tar xvfz; make` and see your answers.

Hints:
We strongly recommend that you explore the `makefile`s we have created, for your convenience. For example, from the top directory:

- `make hw3` will run the code for questions 2-4
- `make package` will create the tar file, for submission
- `make spotless` will clean up all the derived files (*.o, etc)

Make sure that you **exclude** redundant/derived files, in your tar file. Also, it is *your* responsibility to make sure that all necessary files are included in your tarball.

Homework 3 continues…

Question 1: Density Paradox .... [15 points]

On separate page, with '[course-id] [hw#] [question#] [andrew-id] [your-name]'

Grading info: Graded by: Sanjay Chandrasekaran

Problem Description: Suppose you are given (infinite) count of points, uniformly distributed on three axis planes, as shown in Fig. 1. Compute the density at the origin, i.e. $(0, 0, 0)$ point, when we are told that the number of its neighbors $NN(r)$ within radius $r \leq 1$ is

$$NN(1) = 30$$

Assume any norm you want (Euclidean, Manhattan, etc) - your answers should be the same.

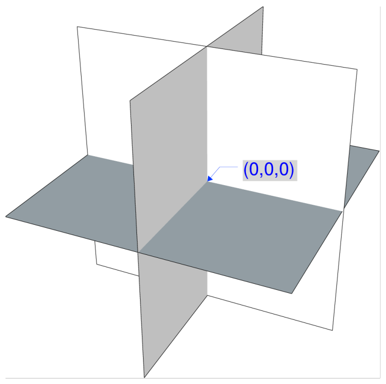


Figure 1: An illustration of axis planes

(a) [**6 points**] Estimate the density for the following radii: r=5; r=10; r=20.

Solution: $D(r = 5) = \frac{NN(5)}{r^3} = \frac{5^2 \times 30}{5^3} = 6$
$D(r = 10) = \frac{NN(10)}{r^3} = \frac{10^2 \times 30}{10^3} = 3$
$D(r = 20) = \frac{NN(20)}{r^3} = \frac{20^2 \times 30}{20^3} = 1.5$
(Assumed L_2 inf-norm, any other norms (e.g. L_2) should also be accepted)

(b) [**4 points**] Based on your estimation, what did you find?

☐ The density is a constant.
☐ The density grows as a power law with the radius.
[✓ **The density shrinks as a power law with the radius.**]
☐ The density fluctuates as the radius increases.

(c) [**5 points**] Ignore Figure 1 - Instead, consider (infinite count of) points, that are uniformly distributed along the line $x_1 = x_2 = x_3 = x_4$ in a four-dimensional space. What is the relationship between the density $D(r)$ at the origin (0,0,0,0), and radius r?

Question 1 continues. . .

☐ $D(r) = C$ ☐ $D(r) \propto r$ ☐ $D(r) \propto r^{1.58}$ ☐ $D(r) \propto r^2$ ☐ $D(r) \propto r^3$
☐ $D(r) \propto r^{-1}$ ☐ $D(r) \propto r^{-1.58}$ ☐ $D(r) \propto r^{-2}$ [✓ $D(r) \propto r^{-3}$]

What to turn in:

- **Answers:** Submit hard copy for the answers.

Homework 3 continues. . .

Question 2: Correlation Integral .... [30 points]

On separate page, with '[course-id] [hw#] [question#] [andrew-id] [your-name]'

Grading info: Graded by: Joey Fernau

Problem Description: The purpose is to show the power of the correlation integral, which helps us find patterns that the points in a dataset may follow. You are given 2 *mystery* datasets, 'A' and 'B', each of which contains 1,000 points in *tar-file package*. Use correlation integral to answer the following questions.

Implementation Details: In the *mystery* datasets, each row contains a point represented by three space-separated floating numbers, i.e. $x\ y\ z$. Implement an algorithm to calculate the correlation integral of those points. We will compare your code against the approximate, but faster, box-counting method is at `http://www.cs.cmu.edu/~christos/SRC/fdnq_h.zip`

(a) [**5 points**] Use Python to implement the naïve $O(N^2)$ algorithm to compute the correlation integral in `./Q2/q2.py`. Include self-pairs and mirror pairs.

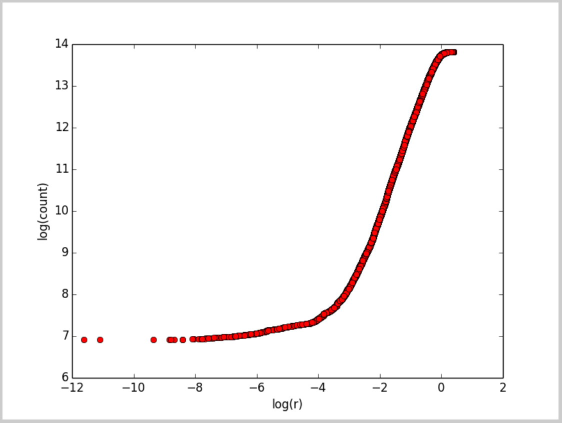
Solution: See solution tar-file.

Grading info:

- *-5 for no code*

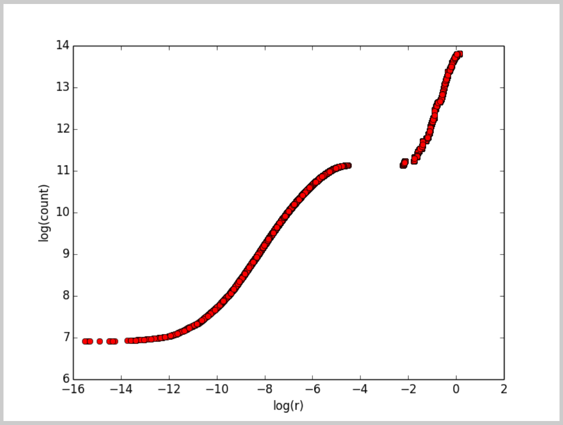
(b) [**5 points**] Plot the correlation integral for datasets 'A' and 'B', respectively. *Hint1*: You may sample (logarithmically, eg., 1,2,4,8, ...) some of the radiuses to avoid plotting too many points. *Hint2*: We recommend the `pyplot` package to plot.

Solution: Dataset A:



Dataset B:

Question 2 continues...



Grading info:

- *-2 for incorrect A*
- *-2 for incorrect B*

(c) For the given datasets, answer the following questions:

i. **[5 points]** Which dataset has more than one cluster? (Notice that a random dataset would form one giant cluster; and so would points on a line, or on a Sierpinski triangle). If both 'A' and 'B' have more than one clusters, mark the one with the most clusters; if they tie, mark 'tie'.

☐ *A* [✓ *B*] ☐ tie

Solution: There is an obvious plateau in the plot for B, which indicates the gap between clusters.

ii. **[5 points]** For the dataset you checked in (i.) (or for 'A', if they tie, with more than 1 clusters), assume the count of points in each cluster is roughly equal; how many clusters are there? Check the nearest one:

☐ 2 ☐ 5 [✓ **15**] ☐ 30 ☐ 60 ☐ 120 ☐ 240 ☐ 480

Solution: From the plot, we can see the number of pairs in each cluster is e^{11}. $(\frac{\#\text{Pairs}}{\#\text{Clusters}})^2 \times \#\text{Clusters} = e^{11}$. Therefore, the number of cluster is $\frac{10^6}{e^{11}} \approx 15$.

iii. **[5 points]** For the dataset (ii.), what is your estimate of the diameter D_c of the clusters? Check the nearest one:

☐ 10^{-1} [✓ 10^{-2}] ☐ 10^{-3} ☐ 10^{-4} ☐ 10^{-5} ☐ 10^{-6}

Solution: From the plot, we can see the diameter is $e^{-4} \approx 10^{-2}$.

iv. **[5 points]** For the dataset in (ii.), what is your estimate for the distance D_n between two neighboring clusters? Check the nearest one:

[✓ 10^{-1}] ☐ 10^{-2} ☐ 10^{-3} ☐ 10^{-4} ☐ 10^{-5} ☐ 10^{-6}

Solution: From the plot, we can see the distance between neighboring cluster is $e^{-2} \approx 10^{-1}$.

Question 2 continues. . .

Grading info:

- *-5 for incorrect part (all or nothing)*

What to turn in:

- **Code:** Submit your code to blackboard, in the `./Q2` directory of your `tar.gz` file.
- **Answers:** Submit hard copy for
  1. your code for (a),
  2. and the answers for all the questions.

Question 3: Substring Editing Distance . . . . [25 points]

On separate page, with '[course-id] [hw#] [question#] [andrew-id] [your-name]'

Grading info: graded by: Mohak Nahta

Problem Description: The goal is to become familiar with the string editing distance, and its dynamic-programming implementation. The intuition behind the *substring* editing distance is that we want to align a *pattern* to a *string* with the minimum editing cost, *ignoring* the cost for the prefix and the suffix. The insertion, deletion, and substitution costs are all $C_i = C_d = C_s = 1$ in their common parts, and 0 in prefix or suffix of the string. For example, for string "discover" and pattern "power", the substring editing distance is 2, and the alignment is shown below:

```
          [prefix]          [suffix]
String:   d i s c * o v e r y
          | | | | |   |     |
Pattern:  * * * * p o w e r *
```

Note that although there are 7 operations in total, the deletions in the prefix part (red) and suffix part (blue), are free. Therefore, we only need to insert "p", and replace "v" with "w". Clearly, the substring editing distance $D(.,.)$ is *asymmetric*: that is, usually $D(a, b) \neq D(b, a)$.

Implementation Details: Implement the calculation of substring editing distance. By giving two parameter, `python q3.py <string> <pattern>` should give the result. Print a single line of the substring editing distance with the format of `Cost:`*your result*. For example,

```
python q3.py discovery power
```

should give

```
Cost : 2
```

Hint: Your code should pass `make sanity-check`.

(a) **[10 points]** Use Python to implement the calculation of substring editing distance in `./Q3/q3.py`. hint Test your code for corner cases (empty string, empty pattern, identical string and pattern, etc). Note that we will check your program against *secret* data (which, of course, we will publish, after the deadline).

Solution: See solution tar-file.

(b) **[15 points]** For each of the following pairs of string and pattern, give the substring editing **matrix** and the substring editing **distance**.

1. String: `saturday`, Pattern: `sunday`

Question 3 continues. . .

2. String: `sunday`, Pattern: `saturday`
3. String: `compensation`, Pattern: `peanuts`

Solution: See Tables (both are considered correct) below.

1. Substring editing distance: 2.

	ϕ	s	u	n	d	a	y
ϕ	0	1	2	3	4	5	6
s	0	0	1	2	3	4	5
a	0	1	1	2	3	3	4
t	0	1	2	2	3	4	4
u	0	1	1	2	3	4	5
r	0	1	2	2	3	4	5
d	0	1	2	3	2	3	4
a	0	1	2	3	3	2	3
y	0	1	2	3	4	3	2

	ϕ	s	u	n	d	a	y
ϕ	0	1	2	3	4	5	6
s	0	0	1	2	3	4	5
a	0	1	1	2	3	3	4
t	0	1	2	2	3	4	4
u	0	1	1	2	3	4	4
r	0	1	2	2	3	4	4
d	0	1	2	3	2	3	4
a	0	1	2	3	3	2	3
y	0	1	2	3	4	3	2

2. Substring editing distance: 3.

	ϕ	s	a	t	u	r	d	a	y
ϕ	0	1	2	3	4	5	6	7	8
s	0	0	1	2	3	4	5	6	7
u	0	1	1	2	2	3	4	5	6
n	0	1	2	2	3	3	4	5	6
d	0	1	2	3	3	4	3	4	5
a	0	1	1	2	3	4	4	3	4
y	0	1	2	2	3	4	5	4	3

3. Substring editing distance: 3.

	ϕ	p	e	a	n	u	t	s
ϕ	0	1	2	3	4	5	6	7
c	0	1	2	3	4	5	6	7
o	0	1	2	3	4	5	6	7
m	0	1	2	3	4	5	6	7
p	0	0	1	2	3	4	5	6
e	0	1	0	1	2	3	4	5
n	0	1	1	1	1	2	3	4
s	0	1	2	2	2	2	3	3
a	0	1	2	2	3	3	3	4
t	0	1	2	3	3	4	3	4
i	0	1	2	3	4	4	4	4
o	0	1	2	3	4	5	5	5
n	0	1	2	3	3	4	5	6

	ϕ	p	e	a	n	u	t	s
ϕ	0	1	2	3	4	5	6	7
c	0	1	2	3	4	5	6	7
o	0	1	2	3	4	5	6	7
m	0	1	2	3	4	5	6	7
p	0	0	1	2	3	4	5	6
e	0	1	0	1	2	3	4	5
n	0	1	1	1	1	2	3	4
s	0	1	2	2	2	2	3	3
a	0	1	2	2	3	3	3	3
t	0	1	2	3	3	4	3	3
i	0	1	2	3	4	4	4	3
o	0	1	2	3	4	5	5	3
n	0	1	2	3	3	4	5	3

Grading info:

- *-5 for each incorrect answer (5 test cases in total)*

Question 3 continues. . .

- *-3 for each incorrect matrix*
- *-3 for wrong tar-file name or andrew-id or structure*

What to turn in:

- **Code:** Submit your code to blackboard, in the `./Q3` directory of your `tar.gz` file.
- **Answers:** Submit hard copy for
  1. your code for (a),
  2. and the matrixes and results for (b).

Homework 3 continues. . .

Question 4: Percolation - phase transitions .... [30 points]

On separate page, with '[course-id] [hw#] [question#] [andrew-id] [your-name]'

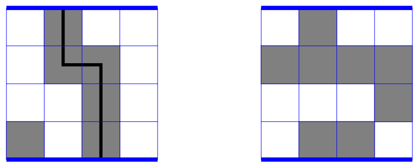
Grading info: graded by: Tanay Varma

Problem Description: The goal is to study the percolation process, and become familiar with the concept of percolation threshold and the counter-intuitive phase-transition phenomenon. As a by-product, this question will also make you familiar with the `networkx` python package, which is excellent for graph/social-network analysis.

The problem considers an $N \times N$ grid, Cells are occupied with probability ('density') p. Two cells are considered as *connected* if they are both occupied, *and* they are neighbors (North, South, East, West; that is, they share a whole side).

A specific grid with free and occupied cells, is defined as *percolated*, if there is a connected path from any occupied cell in the first row to any occupied cell in the last row. See Figure 2(a) for an example, and Figure 2(b) for a counter-example.

The *percolation threshold* or *critical density* p_c is the density that a phase-transition happens - for a finite $N \times N$ grid, we define p_c as the density p at which the probability of a percolated grid, is q=0.5.



(a) percolated (b) non-percolated

Figure 2: Example of 4×4 grids. (a) shows the (only) percolation path (solid black line); (b) has no such path, from bottom to top - the left-to-right path does not count.

Implementation Details: Write a program that sets cells to 'occupied', with probability p, for grids of different sizes, and compute the two measures below (percolation probability, diameter of largest component). Plot a graph depicting the relationship between the density p, and the measure, and report the critical density p_c.

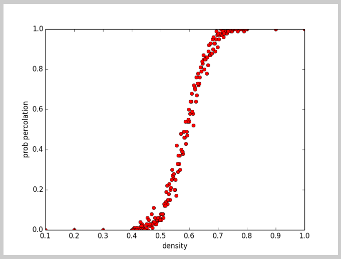
- **Repeat** each experiments 10 times, and take the average - notice the high variability, when we are close to the critical density p_c.
- **Step** the density p by 0.1 in the range 0.1-0.9 (ie., 0.1, 0.2, . . . 0.9), **and** by 0.01 the range 0.4-0.8
- *Hint:* Use the super-useful `networkx` package (already installed on the GHC machines), to do calculations about graphs (diameter, connected components, etc).
- FYI, we will give full points, for *approximately* correct answers - the accurate estimate of p_c is a difficult research problem!

Question 4 continues. . .

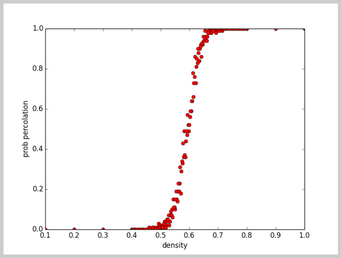
(a) **[15 points]** Use Python to plot a graph of the percolation probability q versus the density p, in `./Q4/q4a.py`. Consider the following grids, and report the critical density p_c (i.e, where the percolation probability q=0.5).

- 16×16
- 32×32
- 64×64

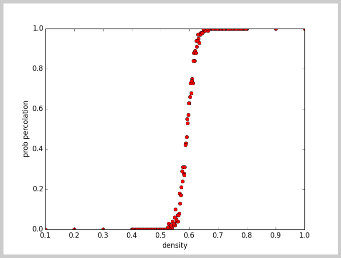
Solution: 16×16:



32×32:



64×64:



Grading info:

- *-5 : No hard copy of code (Only submitted graph)*
- *-5 : For each missing graph*
- *-3 : If estimates are off. I am accepting* 0.60 ± 0.10 *for values of* p_c *for (a) & (b)*
- *-2 : If graphs are correct but no mention of* p_c

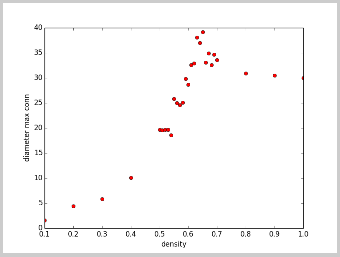
(b) **[15 points]** Use Python to plot a graph of the diameter D of the largest connected component, versus the density p in `./Q4/q4b.py` on the following grids. Report the critical density p_c' where diameter is maximal.

- 16×16
- 32×32

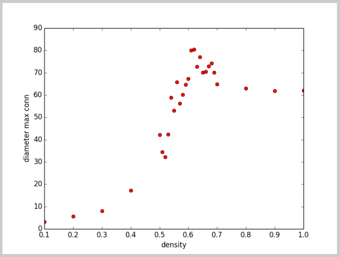
Question 4 continues. . .

- 64×64

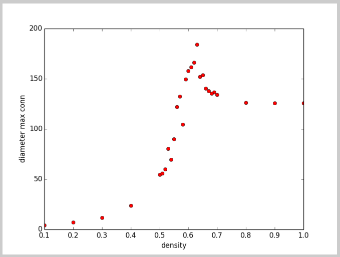
Solution: 16×16:



32×32:



64×64:



Grading info:

- *-5 : No hard copy of code (Only submitted graph)*
- *-5 : For each missing graph*
- *-3 : If estimates are off. I am accepting 0.60 ± 0.10 for values of p_c for (a) & (b)*
- *-2 : If graphs are correct but no mention of p_c*

What to turn in:

- **Code:** Submit your code to blackboard, in the `./Q4` directory of your `tar.gz` file.
- **Answers:** Submit hard copy for
  1. your code for (a) and (b),
  2. and the answers for both questions.