تمرین شبیه سازی سری چهارم
علی قبله 99109971

1. طول هم‌بستگی

در این آزمایش میخواهیم طول همبستگی را در احتمال های متفاوت بررسی کنیم. متوسط اندازه ی تمام خوشه های غیر بینهایت معیاری از طول همبستگی است. انتظار داریم که این مقدار با زیاد شدن احتمال زیاد شود چون اندازه ی خوشه ها بزرگ‌تر میشود و در نقطه بحرانی طول همبستگی به بیشترین حد خود برسد. بعد از آن چون خوشه ها به خوشه بینهایت متصل میشوند این مقدار کاهش می یابد. (منبع: نمونه گزارش کار هدی‌طائب)

این کد از 12 تابع و 5 حلقه در بخش اصلی کد تشکیل شده است.

- تابع percolation check با استفاده الگوریتم هوشن گوپلمن نوشته شده است. که چک میکند آیا تراوش رخ داده است یا خیر.
- تابع real color رنگ واقعی را نمایان می‌کند.
- تابع big_cluster_finder، رنگ و مساحت بزرگترین خوشه غیر بی‌نهایت را به صورت index, max_s، نمایان می‌کند. برای حل مشکل تشخیص چند ریز خوشه متصل به هم به عنوان تراوش، با معرفی یک جمله شرطی و معرفی یک خوشه غیر صفر به جای ریز خوشه های صفر به عنوان بزرگترین خوشه، این مشکل را حل میکنیم.
- توابع cm finder و var finder مراکز جرم و واریانس را در دو راستای x و y بدست می‌آورد. در این سوال و سوال 2 این توابع برای بزرگترین خوشه غیر بی نهایت این مقادیر را به دست می‌آورند.
- تابع value assign یک grid را به سلول های روشن و خاموش تبدیل می‌کند.
- تابع mother به صورت رندوم یک سلول را روشن یا خاموش می‌کند.

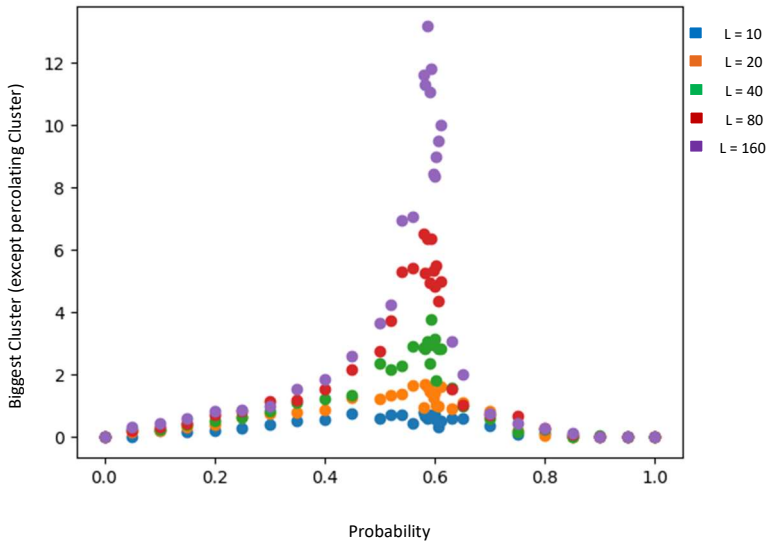
با استفاده از توابع بالا و تشکیل یک صفحه 2 بعدی(با تابع plane init)، می توان با تشکیل 5 حلقه و بدست آوردن نقاط با احتمال مشخص و با تشکیل یک آرایه مساحت به اندازه توان دوم L به اضافه 1، این سوال را حل کرد.

طول در این سوال با توجه به داده های خود سوال L = 10, 20, 40, 80, 160 است.

احتمال به صورت زیر انتخاب شده است:

probability_array = np.array([.0, .05, .1, .15, .20, .25, .30, .35, .4, .45, .50, .52, .54, .56, .58, .583, .587, .59, .593, .597, .60, .603, .607, .61, .63, .65, .70, .75, .8, .85, .9, .95, 1])

برای حل این سوال نیز یک متغیر global به نام counter انتخاب شده است که بتوانیم در توابع از آن استفاده کنیم و میزان آن را تغییر دهیم. (نقاط زیر برای 10 اجرا بدست آمده اند. (Run time: 39 minutes, Intel corei7 gen 9)



طبق نمودار بالا می‌توان مشاهده کرد که در شبکه محدود به یک مقدار ماکسیمم می‌رسیم اما برای شبکه بی نهایت ترمودینامیکی یک نقاط بحرانی خواهیم داشت.

حال می‌بایست نقاط ماکسیمم را برای هر طول یادداشت و با گرفتن میانگین نقطه بحرانی را پیدا کنیم. پس خواهیم داشت:

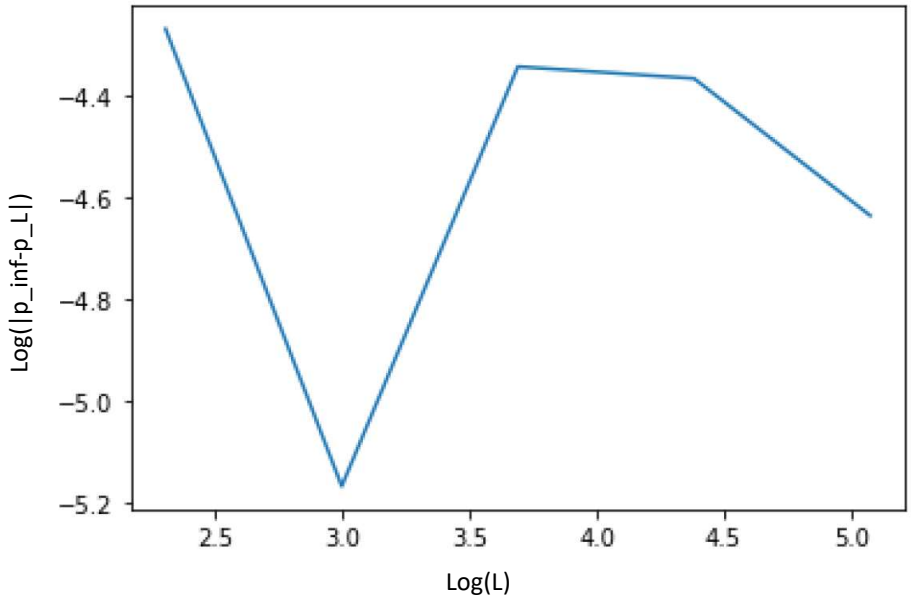
Max x = 0.603(L=10), 0.600(L=20) ,0.590(L=40) ,0.587(L=80) ,0.593(L=160)

- Probability of L = infinity: x = 0.590

در میانگین گیری بالا دو طول 10 و 20 به دلیل کم بودن دیتا های منطقی (دیتا ها با طول زیاد) حذف شده اند. اگر تعداد بیشتر بود (طول ها تا 1000) این دو طول را نیز میتوانستیم در میانگین گیری استفاده کنیم.

2. نمای بحرانی

این سوال دقیقا همانند سوال پیش است و تنها مرحله مورد نظر کشیدن لگاریتم احتمال های طول های همبستگی به لگاریتم طول می‌باشد. نمودار زیر 100 بار اجرا شده است.(Run time: 40 minutes, Intel Corei7 Gen9)



پس از اجرای نرم افزار شیب نمودار و میزان Nou نمایش داده می‌شود که برای 100 بار به صورت زیر است:

best slope(-1/nou) is: 0.44675943692173453

nou is: 2.23834107879222

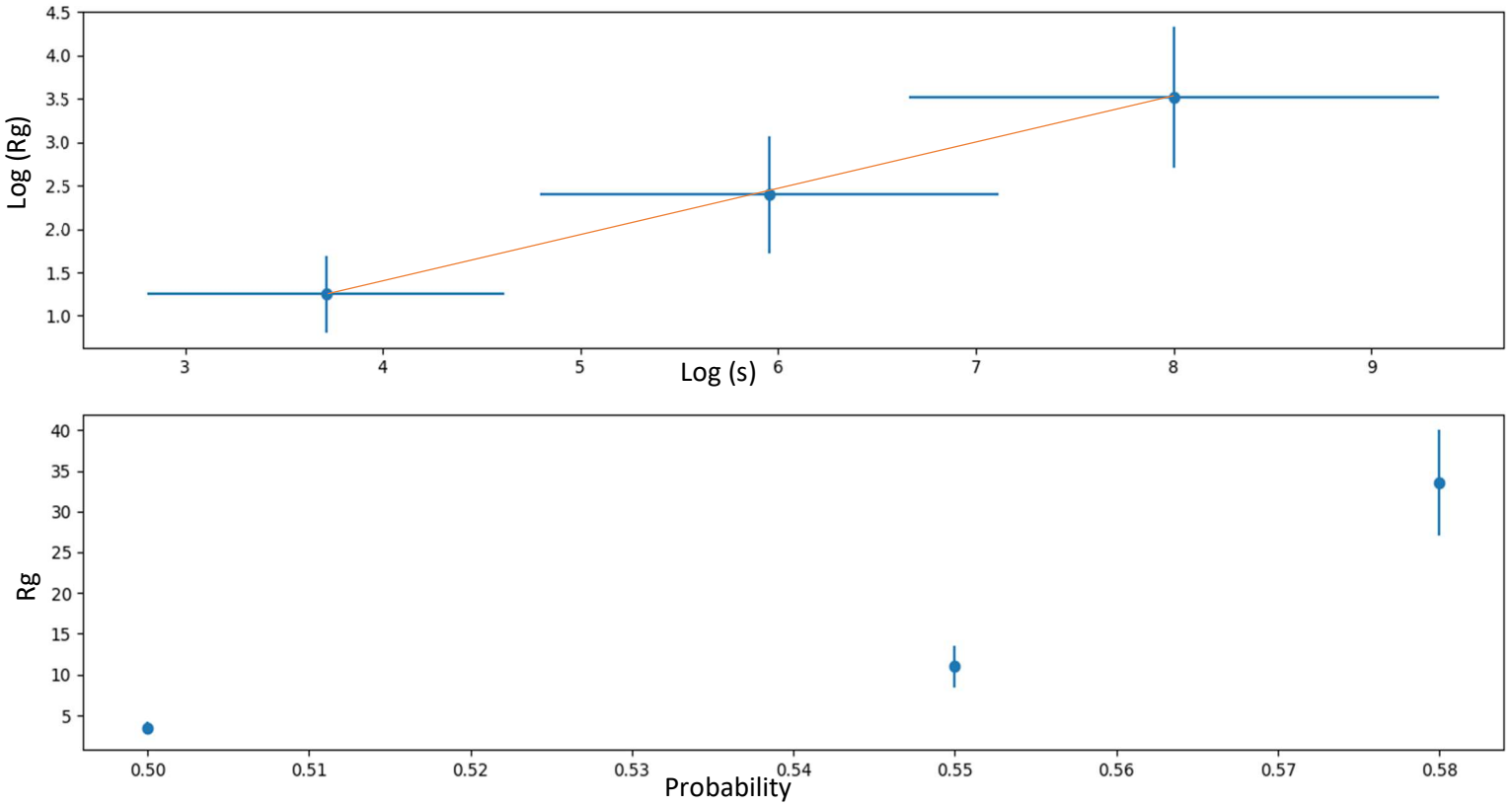
3. بعد جرمی خوشه های تراوش

در این سوال نیز 2 تابع گفته شده در سوال 1 را که توابع پیدا کردن واریانس و بعد جرمی هستند را داریم. دو تابع دیگر را در کنار این دو تابع اضافه می‌کنیم.

- تابع growth، این تابع هربار خوشه را یک قدم بزرگتر می‌کند. در این تابع یک حلقه کلی و چندین شرط وجود دارد که کار این شرط ها چک کردن روشن یا خاموش بودن سلول ها و تشخیص نقاط مرزیست به طوری که -1 به عنوان نقطه خاموش، 0 به عنوان نقطه مشخص نشده، 1 به عنوان نقطه روشن و 2 به عنوان نقطه روشن در منطقه مرزیست.
- تابع evo، این تابع با حضور یک حلقه کلی تمام نقاط مرزی را(که در تابع قبل به عنوان 2 شناخته شده بودند) به توابع روشن تبدیل می‌کند یا به عبارتی مرز را حذف می‌کند.

بدنه اصلی برنامه دارای 2 حلقه جداگانه که هرکدام یک حلقه در خود دارند، می‌باشد. در این دو حلقه میانگین آنسامبلی و شکل خواسته شده به دست می‌آیند.

نمودار اول برای 100 اجرا و اندازه های 301، 401، 1001 و احتمال 0.50، 0.55، 0.58 می باشد و نمودار دوم مقدار Rg بر حسب احتمال را به تصویر می‌کشد.(Run time: 59 minutes, Intel Corei7 Gen9)



حال برای نقاط x و y برای این سوال و شیب آن‌ها خواهیم داشت(برای نمودار اول):

X=3.95,5.80,8.6

Y=1.25,2.25,3.6

⇨ Slope=0.503 (y = 0.5035x - 0.7131)

4. اثبات رابطه زیر:

$$\sigma^2 = < x^2 > - < x >^2 = \frac{4l^2}{\tau} pq\, t.$$

خواهیم داشت:

$$< x_t{}^2 > = < x_{t-\tau}{}^2 > + < a^2 l^2 > + 2 < x_{t-\tau} a l >$$

$$< a^2 > = p + q = 1$$

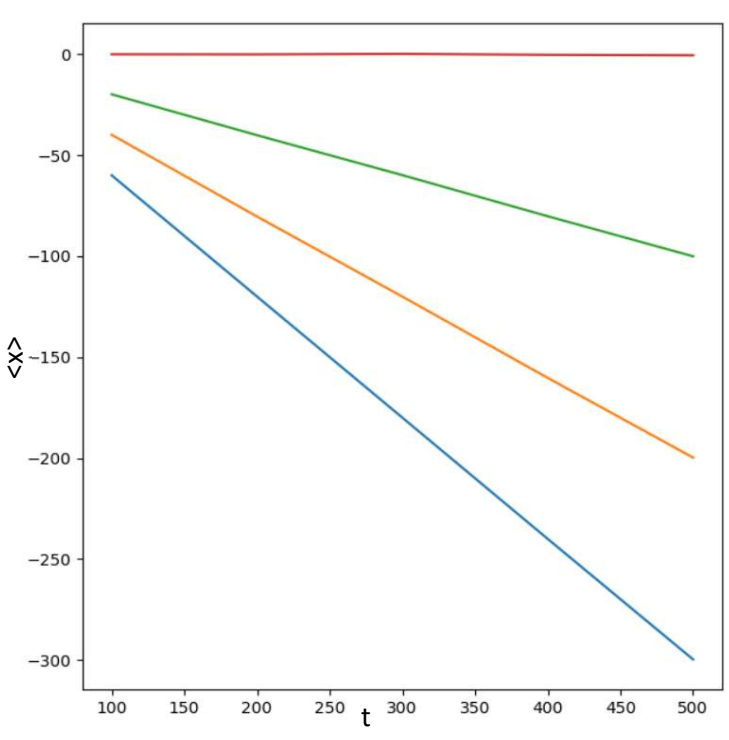
$$\sigma^2 = 4\frac{t}{\tau} l^2 (p^2 + q^2) - 2\frac{t}{\tau} l^2 (p - q)^2$$

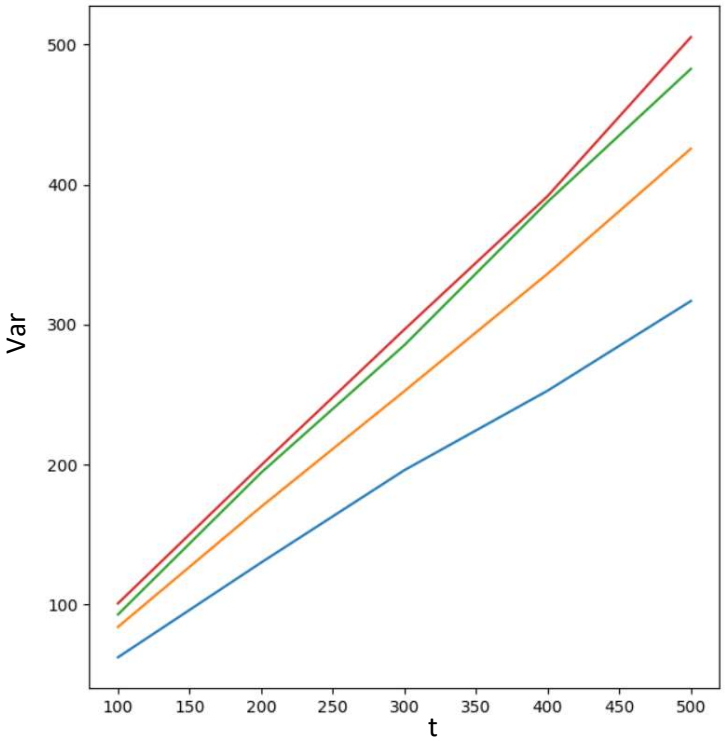
برای جزئیات بیشتر به عکس داخل فایل با نام *Question 4* مراجعه شود.(حل به صورت دستی انجام شده است.)

5. ولگشت ساده

در این برنامه ولگرد یک بعدی را شبیه سازی می‌کنیم. روش کار بسیار ساده است. به تعداد مشخصی که در اول برنامه ذکر شده است(8000 بار) برای احتمال های از 0.2 تا 0.5 (به اندازه های 0.1) برای زمان های 100 تا 500 بار (فواصل 100 تایی)، به طور تصادفی ولگرد را حرکت می‌دهیم و در نهایت با میانگین گیری از ولگرد ها از زمان 1 تا t می توانیم نمودار را رسم کنیم. برای یافتن واریانس نیز با استفاده از فرمول داده شده در کتاب مرجع عمل می‌کنیم.

(Run time: 15 Seconds, Intel Corei7 Gen9)





آبی: احتمال 0.2 نارنجی: احتمال 0.3 سبز: احتمال 0.4 قرمز: احتمال 0.5

شیب های نمودار های مختلف را در اجرای کد می‌توانیم ببینیم:

best slope for line of x_bar with probability= 0.2 is:	-0.5991544999999995
best slope for line of sigma^2 with probability= 0.2 is:	0.6319413750000025
best slope for line of x_bar with probability= 0.3 is:	-0.4005287499999999
best slope for line of sigma^2 with probability= 0.3 is:	0.8499182590624856
best slope for line of x_bar with probability= 0.4 is:	-0.19964275000000015
best slope for line of sigma^2 with probability= 0.4 is:	0.9731647070625026
best slope for line of x_bar with probability= 0.5 is:	-0.0005924999999999999
best slope for line of sigma^2 with probability= 0.5 is:	1.0014622836249998

صحت پاسخ های بالا را می توان با روابط خواسته شده مقایسه کرد.

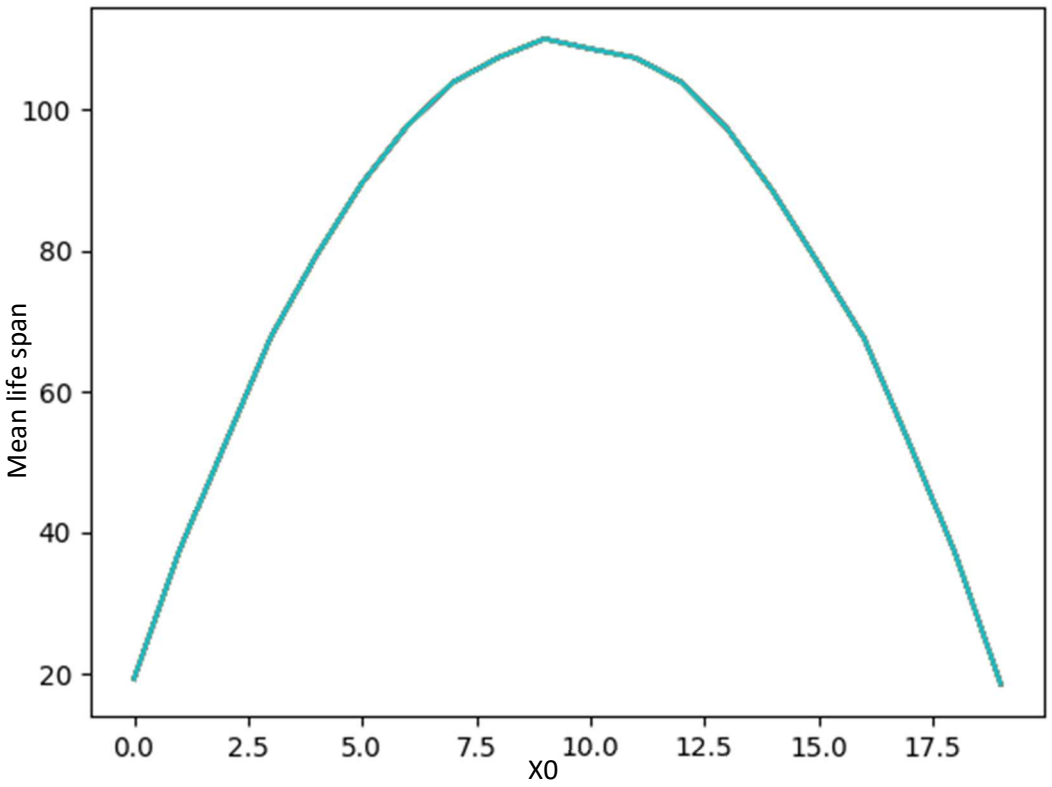
برای 0.2 : واریانس 0.640935861519999872 و میانگین x 0.60087737017499988-

که بسیار به نتایج به دست آمده ما نزدیک هستند. پس صحت حل مشخص می‌شود.

6. ولگشت با تله

در این مسئله همچنان ولگرد یک بعدی را داریم تنها فرق این مسئله آن است که در صورت رسیدن ولگرد به مرز ها در آنجا متوقف می‌شود و به عبارتی نابود می‌شود. مراحل میانگین گیری مانند قبل تکرار می‌شود.

برای تعداد 10000 تا به شکل زیر میرسیم (Run time: 15 Seconds, Intel Corei7 Gen9)



میانگین طول عمر ولگرد ها در انتهای برنامه نشان داده می‌شود و برابر است با: 76.065075

7. حل پرسش قبل با سرشماری

ایده سوال دقیقا همانند بخش قبل است و تنها کافیست یک تعین را وارد الگوریتم کنیم و الگوریتم سوال پیش را تبدیل به یک الگوریتم تعینی کنیم. برای این سوال، برای حفظ سرعت کتابخانه jit را وارد می‌کنیم که این کتابخانه تابع را در c اجرا میکند نه در پایتون!.

میانگین طول عمر ولگرد ها: 76.99985118741331