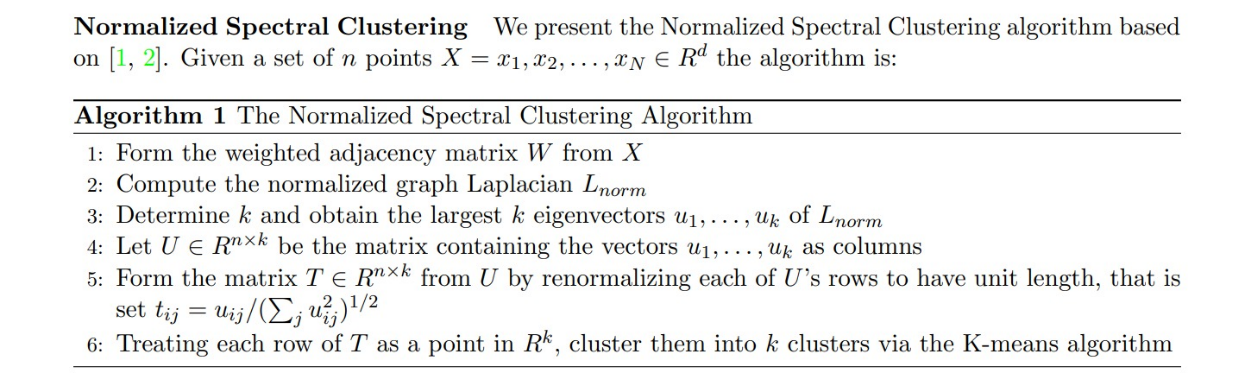
הפרוייקט בסדר כרונולוגי מהרגע שמריצים את קוד הmain שלו:

שלב ראשון:



מקבלים n דאטה פוינטס בתור קלט (וקטורים).

* 1. מרכיבים מטריצת שכנויות בין הנקודות (סימטרית)- נותנים משקל לכל זוג נקודות. במקום הi,j שמים משקל שהוא אינדיקציה לכמה הן רחוקות. קרוב = משקל קטן.
  2. יוצרים מטריצה אלכסונית. בתא הi,i שמים את סכום השורה הi ממטריצת השכנויות.
  3. מפעילים על המטריצה האלכסונית העלאה בחזרת -0.5, כל איבר בה הופך להיות אחד חלקי השורש של עצמו.
  4. מוציאים מהמטריצה האלכסונית וממטריצת השכוניות את מטריצת הלפלסיאן (Lnorm).

למה מעוניינים בה? בגלל שכל הערכים העצמיים שלה יהיה אי שליליים ולא מרוכבים (ממשיים).

* 1. מוצאים ערכים ווקטורים עצמיים: לוקחים את הלפסיאן ומריצים עליה את אלג' jacobi. אלג' יעקובי מקבל את מטריצה A בתור קלט ההתחלתי (ז"א A= מטריצת הלפסיאן בתור הקלט שנשלח לו). על מטריצה A הזו: \*להבין איך עשינו יעקובי ולחזור
  2. משיגים מהפעלת יעקובי את המטריצה A הסופית וגם את המטריצה V, שזו הכפלה של מטריצות הסיבוב (לראות את פרטי יעקובי). האלכסון של A היא הערכים העצמיים של Lnorm ומטריצה V תהיה בעלת כל הוקטורים העצמיים של Lnorm. כל התכלית של יעקובי זה למצוא את הערכים העצמיים והוקטורים העצמיים של מטריצה.

התוצאה: n ערכים עצמיים וn וקטורים עצמיים (בהנחה שיש n דאטה פוינטס)

1.7**(אייגןגאפ יוריסטיק)**  ממיינים את הערכים העצמיים בסדר עולה ועוברים עליהם כדי למצוא את הקפיצה הכי גבוהה בין שני ערכים עצמיים בהפרש. ברגע שמוצאים אותה (ההפרש המקסימלי בין שני ערכים עצמיים גבוהים) אז איפה שהקפיצה התבצעה בסדר הממויין של הערכים – זה k (כלומר האינדקס של האיבר הראשון מבינהם). המטרה כאן הייתה למצוא את K, שאיתו נעבור לשלב הבא.

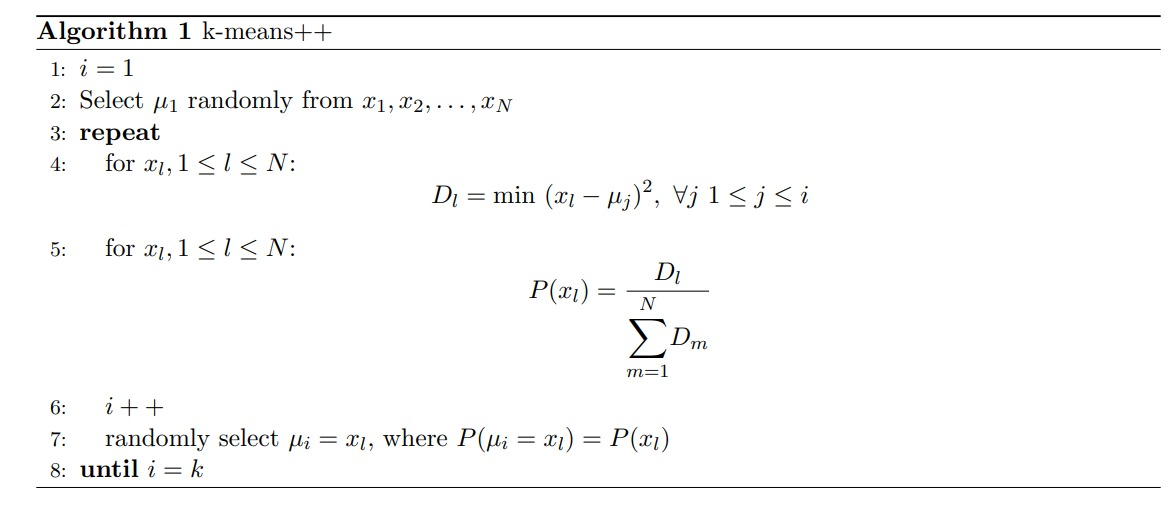
1.8 לוקחים את k האייגן וקטורס הכי קטנים (אצל אביב שינו את זה לגדלים) ולוקחים את ה k וקטורים עצמיים התואמים אליהם. משבצים בעמודות את k הוקטורים העצמיים האלה במטריצה חדשה – U, ממימד n על k.

1.9 הופכים את U למטריצה חדשה T- שזה פשוט U מנורמלת לפי ההסבר שבפסאודו קוד.

1.10 לוקחים את T שהיא מטריצה עם n שורות וk עמודות ומסתכלים על השורות. כל שורה היא מבחינתו דאטה פוינט, אז בעצם יש עכשיו n דאטה פוינטס וכל דאטה פוינט הוא ממימד k, והוא הולך להיות הקלט ההתחלתי בתור דאטה פוינטס של kmeans++ בשלב הבא.

שלב שני:

Initialize(במילים אחרות- kmeans ++)



מה קורה בו:

1. לולאת while שרצה k פעמים. בכל איטרציה i:

1.1אם זו האיטרציה הראשונה: בוחרים את הצטנרוראיד רנדומלית מתוך n הדאטה פוינטס.

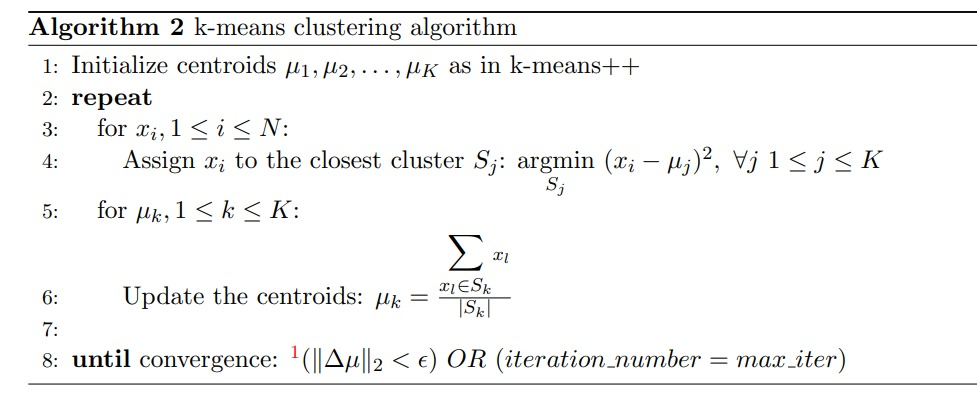
אחרת (i>=2) :קובעים מרחק מינימלי לכל אחד מהדאטה פוינטס מהi-1 צנטרואידים קודמים שכבר צברנו-קוראים למרחק dl.

לכל דאטה פוינט קובעים הסתברות בחירה להיות הצנטרואיד הבא שנבחר- probability שקובעים אותה לפי הdl שלו, כדי שבאיטרציה הבאה יהיה לו סיכוי אמנם רנדומי, אבל כן מוטה יותר לפי המרחק dl המינימלי שלו (כלומר המרחק המינימלי של הנקודה מצנטרואיד אחר). ככל שלנקודה כזו יש dl מינימלי נמוך יותר (מרחק נמוך יותר מצנטרואיד אחר כלשהו מהצנטרואידים שנצברו)- נגדיר את הסיכוי p שלה (להבחר בתור צנטרואיד באיטרציה הבאה) להיות גדול יותר. \* הערה- המרחקים הם ריבועיים- להסתכל בפרטים אחרכך.

* 1. לפני האיטרציה הבאה של הwhile עושים את הבחירה לפי הסיכויים שהגדרנו, מי יהיה הצנטרואיד הבא שייכנס לרשימת הצנטראודים.

1. אחרי k איטרציות, כלומר אחרי שאספנו k צנטראוידים שונים מתוך הדאטה פוינטס, אנחנו מחזירים אותם בתור k הצנטראודיים ההתחלתיים (הם עדיין נקודות אמיתיות מתוך הדאטה פוינטס) לאלגוריתם שהולך לקבל אותם- kmeans הרגיל (כלומר מגיעים לפה אם goal = spkmeans)

כל זה(kmeans++) היה בעצם שלב 1 פה:

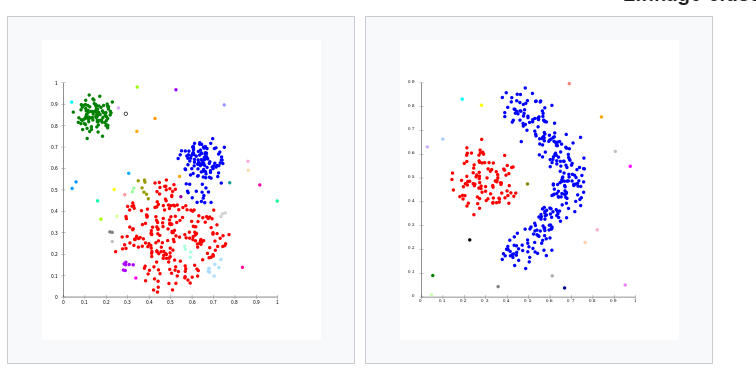


שלב 3:

K means הרגיל (לא++), שזה בדיוק מתחיל בשלב 2 איפה שכתוב repeat:

מקבלים מהשלב הקודם k צנטרואידים התחלתיים.  
מקבלים דאטה פוינטס (וקטורים ממימד כלשהו, לא ידוע איזה).

צריך לבצע Normalized spectral clstering עליהם. ז"א שעושים עליו שזה פעולות מתמטיות כל הוקטורים האלה- בעצם מחלקים את



K means הרגיל:

כל אחת מהנקודות האלה- זה דאטה פוינט

כל צבע זה קלאסטר. בהתחלה מתחילים עם k קלאסטרים (3) . בוחרים בהתחלה 3 נקודות אקראיות. הקלאסטרים שמתחילים איתם את האלגוריתם הם אקראיים לגמרי. בוחרים k נקודות מתוך כל הn דאטה פוינטס (הקלט) שזה היה במטלה הראשונה.

ואז בככל איטרציה (מהשנייה ואילך):

בוחרים לכל קלאסטר צנטרואיד- נקודה ממוצעת של כל הקלאסטר.

מנתקים את הצבעים מכל הנקודות שוב, ואז עושים מיון מחדש לקלאסטרים אבל הפעם לפי k הצנטראודים.

חוזרים על התהליך שוב ושוב ברגע שיש convergance (כשהנורמה האוקלידית של כל אחד מהצנטרואידים לא זזה ביותר מאפסילון כלומר כשבפועל הצנטרואידים נשארו בין שתי איטרציות באותו המקום).

בסופו של דבר אתה מחזיר k צנטרואידים