**תרגיל כיתה 12 – Principle Component Analysis (PCA)**

טעמנו לאורך חלקו השני של הקורס את הקושי בהתמודדות עם מידע במימד גבוה. PCA (Principal Components Analysis – ניתוח גורמים ראשיים) היא שיטה סטטיסטית למציאת ייצוג ממימד נמוך למידע רב מימדי.

מטרת השיטה היא להוריד את המימד של המידע למימד נמוך יותר תוך שמירה על רוב המידע באופן המאפשר חשיפת קשרים בין המשתנים השונים אשר נחקרים (בפועל התוצר של PCA יכול להיות באותו מימד כמו המידע המקורי, אך לרוב ניתן להסתפק בגורמים הראשיים הראשונים כי אלו מכילים את רובה הגדול של האינפורמציה).

נניח שיש לנו מידע במימד גבוה – למשל, אם עבור מטופלים בחדר מיון נתונים 10 מדדים, אזי המימד של המידע הוא 10. אפשר לדמיין כל דגימה כנקודה במרחב 10 מימדי. אלגוריתם PCA יצור מערכת צירים חדשה, כך שהציר הראשון תופס את רוב **השונות** של הנתונים, הציר השני תופס את רוב השונות של המידע במרחב המאונך לציר הראשון, וכן הלאה. הצירים במערכת הצירים החדשה נקראים "גורמים ראשיים" (Principle Components), ומכאן שמו של האלגוריתם.

באופן אינטואיטיבי נוכל להגיד שהאלגוריתם מסובב את מערכת הצירים הרב מימדית כך שבמערכת הצירים החדשה רוב השונות במידע נמצאת ומוצגת בציר הראשון, שונות קטנה יותר בציר השני וכן הלאה. כשהמידע מוצג כך, נוכל להחליט להסתפק במספר מועט של מימדים (למשל שניים או שלושה) המכילים את רוב אינפורמציה. כך נוכל להציג גרפית את המידע, וגם נוכל מהתבוננות בגורמים הראשיים להבין את הקשרים בין המשתנים השונים (למשל גובה ומשקל, בדוגמא לעיל).

כל ציר ב-PCA יהיה קומבינציה לינארית של המימדים אשר בנתונים שלנו. לדוגמא:

$$PC1=\left[h\right]+\left[0.2⋅w\right]+\left[0⋅BMI\right]+\left[0.6⋅WBC\right]+…$$

$$PC2=[0.4⋅h]+[0⋅w]+[0⋅BMI]+[0.2⋅WBC]+…$$

כל ציר PC מתאר אחוז מסויים מהשונות של הנתונים, והאלגוריתם מסדר את הצירים כך ש-PC1 מתאר את אחוז השונות הגבוה ביותר, PC2 השני הגבוה ביותר וכו'.

המשקל של כל אחד מהמימדים בכל אחד מצירי ה-PC נקרא loading.

הקואורדינטות החדשות של כל דוגמא במרחב ה-PCA נקראות scores.

**איך עובד האלגוריתם?**

1. מרכוז (centering) של הנתונים כך שהממוצע של כל משתנה בכל מימד הוא אפס.
שימו לב שהמרכוז לא משנה את איך שהנתונים מפוזרים על הצירים.
2. נחפש ציר (שעובר דרך הראשית) וייתן לנו את השונות הגבוהה ביותר האפשרית. לכל קו שעובר דרך הנקודות מחשבים את סכום ריבועי המרחקים מכל נקודה לקו. הציר שמצמצם את הסכום הזה בצורה המשמעותית ביותר יהיה PC1.
3. כעת, מתוך כל הצירים שניצבים (אנכיים) לציר הראשון, נחפש את הציר שנותן לנו את השונות המקסימלית. זהו PC2.
4. נמשיך בצורה הזו עד שנסיים.
5. לאחר שנמצא את כל ה-PCs, נעביר את הנתונים למערכת הצירים החדשה
6. ניתן להזניח חלק מהמימדים שהשונות המוסברת בהן נמוכה.

**דוגמא בדו-מימד**

על מנת לקבל הרגשה ואינטואיציה לגבי משמעות הדברים נדגים תחילה על ידי דוגמא טריוויאלית בשני מימדים. נטען את קובץ הנתונים ונציג אותם:

Data <- read.csv("TA13.csv", header = F)

plot(Data, xlim = c(-10,10), ylim = c(-10,10), col = 'blue')



מהם כיווניהם של הגורמים הראשיים הראשון והשני?

אפשר לראות בעיניים את הכיוון שתופס את רוב השונות של הנתונים:



הקו האדום מייצג את הכיוון בו רוב השונות של הנתונים מיוצגת – כלומר, הקו הזה מייצג את הנתונים בדרך הטובה ביותר. הקו המאונך לו מייצג את הציר בו שאר השונות בנתונים מיוצגת. זוהי פחות שונות מהציר הראשון, והוא נמצא בצורה מאונכת לקו הראשון.

זה בעצם מה שעושה אלגוריתם ה-PCA – מוצא את הקומבינציה הלינארית של המימדים אשר נותנת לנו את השונות הגבוהה ביותר בנתונים, ולאחר מכן מחפשת את הקומבינציה הבאה המאונכת (אורתוגונלית) לקומבינציה הקודמת. ככל שהאלגוריתם מתקדם כך כמות השונות שנשארת בנתונים יורדת (כי היא כבר יוצגה), ולכן כל ציר PC ייצג אחוז שונות הולכת ויורדת.

כעת נשתמש באלגוריתם ה-PCA על מנת לחשב את זה "רשמית":

p <- prcomp(Data, scale. = F)

הפונקצייה מקבלת את המדדים הבאים:

* מטריצה (במקרה שלנו Data). יכולה להיות רק המספרים, יכולים להיות בה גם משתנים שלא נרצה להכניס ל-PCA, אך נרצה להשתמש בהם לאפיון לאחר מכן.
* scale. – מקבלת TRUE/FALSE, האם לנרמל את הנתונים לפני חישוב ה-PCA. בשלב זה בשיעור, נבחר FALSE.

הפונקציה תחזיר לנו אובייקט מסוג list עליו למדנו בשיעור הקודם – ואשר יכול להכיל אובייקטים מסוגים שונים (כגון character vectors, data frames וכו'). מה האובייקט מכיל?

* Rotation – מטריצת loadings בגודל NxN כאשר N – מספר המשתנים
* x – מכיל את המיקום על הצירים של כל אחת מהדוגמאות.

נציג את אחוז השונות המוסברת על ידי כל אחד מהגורמים הראשיים (PCs):

summary(p)



כלומר, הציר הראשון (PC1 או Dim.1) תופס 87.15% מהשונות בעוד שהציר השני תופס 12.85%. יחד הם מייצגים 100% מהשונות בנתונים.

ניתן לראות זאת גם ע"י גרף Scree עליו למדנו בשיעורים הקודמים:

fviz\_screeplot(p)



כעת נסתכל על גרף ה-PCA:

אנו רוצים גרף אשר מתאר איך כל דוגמא נראית במרחב ה-PCA ולכן נשתמש בפונקציה אשר יוצרת גרף PCA לכל דוגמא (individual = ind):

fviz\_pca\_ind(p, geom = "point")



כך נראה גרף ה-PCA. אך הוא לא נראה כלל כמו הגרף שראינו קודם שגם הוא היה בדו מימד. למה? זה עניין של צירים:



כעת ניתן לראות כי גרף ה-PCA הוא אותו הגרף הדו-מימדי כמו קודם, רק מסובב על מנת שציר ה-X (PC1) יתפוס את מירב השונות.

ניתן גם לראות את הקורולציה של כל משתנה (variable) עם כל PC:

fviz\_pca\_var(p)

ניתן לראות כי המשתנה V2 נמצא בקורולציה בעיקר עם PC2 מכיוון שהחץ שלו נמצא בעיקר על ציר ה-y. מכיוון שהחץ מופנה למטה ניתן לומר כי ככל שערך V2 עולה, כך גם מיקום הדוגמא על PC2 נמוך יותר.

כך עובד האלגוריתם גם עבור רב-מימד – הדוגמא הזו הייתה קצת טריוויאלית, כי מלכתחילה הנתונים שלנו היו ממימד נמוך ולא היה צורך להפחית מימדים. PCA רק סובב את מערכת הצירים כך שהצירים יהיו לפי הגורמים הראשיים (PCs). במקרה של נתונים רב-מימדיים הסיבוב יהיה במרחב רב מימדי. נוכל לבחור מספר מועט של מימדים (גורמים) שפורסים את רוב השונות של הנתונים, ולהסתפק בהיטל של הנתונים על הגורמים הללו.

**הערה:** שימו לב כי PCA איננו מנרמל את העמודות. לעיתים הסקאלות אינן זהות במימדים השונים (למשל: לחץ דם, משקל, חום גוף), ובמקרה כזה נצטרך לנרמל את הנתונים לפני ביצוע PCA כדי למנוע "השתלטות" מלאכותית של מימד אחד על כל השונות של הנתונים. על מנת לבצע את הנרמול כאשר אנו קוראים לפונקציה PCA יש צורך לשנות את scale. ל-TRUE.

עניין הנרמול הוא עניין חשוב, אך עם זאת ישנם מקרים בהם נרצה לעבוד עם המידע הגולמי ללא הנרמול ולראות את תוצאות לפי הסקאלות המקוריות של המידע. נסו בעצמכם לעבוד עם וללא נרמול ובידקו את השוני.

**תרגיל במימד גבוה – לימפומה**

1. טענו את מאגר הנתונים של חולי הלימפומה. זוהי מטריצה בגודל 130x29 כאשר העמודה הראשונה מתארת האם הטיפול בחולה הצליח או לא (TRUE/FALSE) וערכי 28 גנים. כלומר, עבור כל מטופל יש מידע 29 מימדי – כל נבדק מיוצג ע"י נקודה במרחב 29 מימדי).

lymphoma <- read.csv("lymphoma.csv", header = T)

1. בצעו PCA וציירו את הגרף המתאים. כמה שונות תופסים שני הצירים הראשונים?

p <- prcomp(lymphoma[,-1], scale. = T)

fviz\_pca\_ind(p, geom = "point")

למה נרמלנו את המשתנים? כפי השסברנו קודם, חשוב לנרמל את הדאטה לפני ביצוע PCA.



ניתן לראות כי PC1 מייצג 33.5% מהשונות ו-PC2 מייצג 7.7% מהשונות. ניתן לראות זאת גם בגרף Scree:

fviz\_screeplot(p)

1. הביטוי של איזה גן או גנים יורד ככל שנמצאים יותר ימינה על PC1?
נפתור זאת בשתי דרכים. ראשית נבין מהי התרומה של כל גן ל-PC1. לשם כך נשתמש ב-loadings שחושבו, אשר נמצאים בתוך האובייקט p:

loadings = p$rotation

barplot(loadings[,1], las=2)



ניתן לראות שגן 9 וגן 28 שניהם בעל loading שלילי על PC1 ולכן ככל שהקואורדינטות של PC1 הן יותר בצד החיובי, כך הערך של שני הגנים הנ"ל נמוך יותר.

נראה זאת גם בעזרת גרף משתנים:

fviz\_pca\_var(p)



1. מהם ה-scores של כל מטופל?

ה-scores הם בעצם הקואורדינטות של כל מטופל ברב מימד. ניתן לקבל את הקואורדינטות כך (הנתונים מוצגים עבור 5 המטופלים הראשונים):

p$x



1. הציגו את כל הדגימות על גרף PCA ואת הנקודות הציגו בצבע שונה עבור מטופלים עבורם הטיפול הצליח ונכשל.

fviz\_pca\_ind(p, geom = "point", col.ind = lymphoma$Result, pointsize = 2)



ניתן לראות שיש הפרדה לא רעה על PC1. מה זה אומר? זה אומר שרוב השונות בנתונים מייצגת את ההבדל בין מטופלים שהטיפול עבורם הצליח לעומת מטופלים שהטיפול עבורם לא הצליח. במילים אחרות, ביטוי הגנים שאנו בודקים, ואשר משפיעים בצורה משמעותית על PC1 גורמים (או חוזים) האם טיפול יצליח או לא.

1. מה יקרה אם נשתמש ב-2PC ו-3PC?

fviz\_pca\_ind(p, geom = "point", axes = c(2,3),col.ind = lymphoma$Result, pointsize = 2)

כעת ההפרדה בנתונים הרבה פחות ברורה. מדוע? מערכת הקורדינטות שנבחרה מייצגת פחות שונות בדאטה.

1. מה קורה כאשר לא מנרמלים את הנתונים לפני?
נאפיין את הנתונים לפני הנרמול:

p <- prcomp(lymphoma[,-1], scale. = F)

fviz\_pca\_ind(p, geom = "point")

loadings = p$rotation

barplot(loadings[,1], las=2)

fviz\_screeplot(p)

fviz\_pca\_ind(p, geom = "point", col.ind = lymphoma$Result, pointsize = 2)



שימו לב שכעת הגורם הראשי הראשון כמעט מקביל למימד הראשון בדאטה. כלומר, המימד עם השונות הכי גדולה 'השתלט' על הגורם הראשי הראשון.



קיבלנו גרפים שונים לחלוטין שלא באמת מסבירים את השונות האמיתית בדאטה. נזכור ש-PCA הוא אלגוריתם שמוצא מערכת קורדינטות חדשה שתמקסם את השונות בדאטה. כאשר הדאטה לא מנורמל משתנה אחד (שמבוטא ביחידות גבוהות) יכול "להשתלט" על הגדרה זו ולפגוע באיכות מערכת הקורדינטות החדשה שנבחר.