

2023



המחלקה למתמטיקה
Department of Mathematics

פרויקט מסכם לתואר בוגר במדעים (B.SC)
במתמטיקה שימושית

זיהוי פנים

אלה ישראלביץ

FACE RECOGNITION

ELLA ISRAELVICH

ADVISOR:
LAVI KARP

מנחה:
פרופ' לביא קרפ

KARMIEL

כרמיאל

תוכן עניינים

3.....	מבוא
4.....	סקירה ספרותית
6.....	זיהוי פנים
7.....	רקע מתמטי באלגברה לינארית
7.....	1. ערכים עצמיים ווקטורים עצמיים
7.....	2. מכפלה סקלרית
7.....	3. נורמה של ווקטור
7.....	4. אורתוגונליות ואורתונורמליות
8.....	5. נורמה של מטריצה
8.....	תכונות הקשורות לנורמה של מטריצה:
9.....	פירוק SVD
9.....	רקע היסטורי
9.....	הגדרה פורמלית
10.....	פירוק SVD מנקודת מבט גיאומטרית
11.....	פירוק SVD מצומצם
12.....	משפט הקיום
12.....	הוכחת משפט הקיום
14.....	חישוב נורמות ע"י פירוק SVD
14.....	דוגמא לפירוק SVD של מטריצה A ממטית
16.....	Principal Component Analysis
16.....	ממוצע ומטריצת השונות המשותפת
17.....	מטריצת השונות המשותפת
18.....	דוגמא למציאת מטריצת שונות משותפת
19.....	תהליך PCA
20.....	דוגמא למציאת רכיבים עיקריים
21.....	דוגמא לחישוב אחוזי השונות של הנתונים
22.....	SVD ושונות משותפת
23.....	שימוש במטריצת השונות המשותפת לחישוב ההסתברות שמדידה נופלת בתוך אזור הפנים הממוצעות
25.....	דוגמא לחישוב הסתברות שמדידה נופלת בתוך אזור הפנים הממוצעות
27.....	ייצוג תמונה במטריצה
28.....	Eigenfaces (פנים עצמיות) - זיהוי פנים ע"י שימוש ב- SVD ו- PCA
29.....	תיאור האלגוריתם
30.....	שימוש בפנים עצמיות
30.....	שיטות נוספות לזיהוי פנים

30.....	שיטות התאמה הוליסטיות
31.....	שיטות גיאומטריות
31.....	שיטות המרקם המקומי
31.....	שיטות למידה עמוקה
32.....	השוני בין שיטת ניתוח רכיבים עיקריים PCA לשיטת ניתוח לינארי של הדסקרימיננטה LDA
32.....	השוני בין שיטת ניתוח רכיבים עיקריים PCA לשיטת ניתוח רכיבים עצמיים ICA
32.....	ICA, LDA ו-PCA
33.....	מימוש זיהוי פנים ב-python
33.....	חלק מספר 1- ייצוג מערך האימון
34.....	פלט חלק 1
35.....	חלק מספר 2- הצגת תמונות הפנים עם תנאי תאורה שונים
36.....	פלט חלק 2
40.....	חלק מספר 3 - חישוב הפנים העצמיות (Eigenfaces)
41.....	פלט חלק 3
42.....	חלק מספר 4 – הטלת תמונות פנים על תת המרחב הנפרש ע"י הפנים העצמיות
43.....	פלט חלק 4
45.....	בביבליוגרפיה

מבוא

פרויקט זה עוסק בזיהוי פנים. זיהוי פנים הוא דרך לזהות או לאשר את זהותו של אדם באמצעות פניו. מערכות לזיהוי פנים יכולות לזהות פנים אנושיות בתמונות, בסרטונים, בזמן אמת, לקבוע אם הפנים בתמונות שונות שייכות לאותו אדם, או לחפש פנים בין אוסף גדול של תמונות קיימות. למערכות אלו יש שימוש בימנו, מערכות אבטחה ביומטריות משתמשות בזיהוי פנים על מנת לזהות באופן ייחודי אנשים במהלך כניסת משתמש או התחברות, כמו כן לחזק את אימות המשתמש. מכשירים ניידים ואישיים גם משתמשים בדרך כלל בטכנולוגיות ניתוח הפנים לאבטחתם.

הפרויקט יעסוק בשיטה לזיהוי הפנים על ידי פירוק SVD, Singular Value Decomposition, ועל ידי PCA, Principal Components Analysis. במהלך הפרויקט אציג את השיטות הללו, אסביר עליהן ואביא דוגמאות מספריות ליישומן. כמו כן, אציג מימוש של השיטה לזיהוי הפנים בשפת התכנות python, כולל תוצאות ויזואליות.

סקירה ספרותית

סקר הספרות בעבודה זו עוסק בשיטות המרכזיות שבהן התמקדנו בתהליך זיהוי הפנים. השיטות העיקריות בהן אנו משתמשים לתהליך זיהוי הפנים הן פירוק לערכים סינגולריים (SVD) וניתוח רכיבים עיקריים (PCA). בעזרת כלים אלו נמצא את הייצוג של הפנים העצמיות Eigenfaces לשם זיהוי הפנים.

על מנת להשתמש בשיטות אלו, ולצורך הבנתן יש צורך ברקע מתמטי נרחב אשר רובו נלקח מאלגברה לינארית [1] (C. Lay). בתהליך זה נעשה שימוש רב במבנה המטריציוני, לשם הצגת תמונות הפנים השונות במאגר במבנה מתמטי ברור. לכן, צריך הבנה במושגים כגון: ערכים עצמיים ווקטורים עצמיים, נורמה של וקטור ונורמה של מטריצה, אורתוגונליות ועוד. בנוסף, יש צורך ברקע מתחום הסטטיסטיקה על מנת להבין לעומק את תהליך מציאת מטריצה השונות המשותפת בתהליך זיהוי הפנים. מושגים כמו תוחלת, ממוצע המדגם, שונות משותפת, ממוצע המדגם, מטריצת השונות, קורלציה וכו' [10] (M. Ross).

כדי לבצע זיהוי פנים יש צורך בהבנה כי לפנים האנושיות יש שונות המשתנה בהתאם לתנאים מסויימים. למשל, לתנאי התאורה בעת הצילום יש השפעה גדולה על הפנים, להבעות הפנים השונות בהתאם למצבי הרוח המשתנים. שונות זו משפיעה על מערכת לזיהוי הפנים ומציבה בפניה אתגרים. מערכות זיהוי פנים באופן כללי נוטות לכלול שלב עיבוד מראש בו התמונה מנורמלת ביחס לווריאציות שהוזכרו לעיל. כלומר, התמונה הופכת לכמה גדלים סטנדרטיים, מבחינת מיקום וכו'. גודל ומיקום הם מאפיינים שקל להתמודד איתם, אבל האחרים בעייתיים לשם כך פותחו מערכות אוטונומיות רבות בתחום, כאשר ביניהם הפנים העצמיות Eigenfaces [9] (Magaia, M. Herbst). בנוסף, הטיפול בתמונות הפנים בגוויי אפור מאפשר לנו לעקוף את בעיית התאורה המשתנה.

פירוק לערכים סינגולריים SVD - Singular value decomposition, היא שיטה חשובה לפירוק מטריצות. זהו אחד הגורמים החשובים והשימושים ביותר האלגברה לינארית [9] (Magaia, M. Herbst). היא מהווה בסיס לשיטות רבות וחשובות ומספקת לנו פירוק יציב מבחינה מספרית אשר מובטח להתקיים. פירוק SVD משמש להשגת דרגה נמוכה ומקרובת של מטריצה, מציאת דרגה של מטריצה ע"י הערכים הסינגולריים, מציאת מרחב האפס, דרך למציאת יציבות של בעיות מתמטיות, ולהיפוך של מטריצות לא ריבועיות (Pseudo-Inverse), על מנת למצוא את הפתרון למערכת משוואות לינארית $A\vec{x} = \vec{b}$ [14] (ויקיפדיה).

השימוש החשוב בשביל עבודה זו, הוא אלגוריתם הבסיס של ניתוח רכיבים עיקריים (PCA). נתונים ממד גבוהה מפורקים לגורמים סטטיסטיים תיאוריים ביותר ונותן חישוב ללא צורך בחישוב ישיר של מטריצת השונות המשותפת [9] (Magaia, M. Herbst). PCA ו-SVD הוחלו על מגוון רחב של בעיות בהנדסה. "לשיטה זו שימושים רבים בעיבוד אותות ובסטטיסטיקה" [14] (ויקיפדיה).

פירוק SVD מספק בסיס המותאם לנתונים ספציפיים. בתחומים רבים מערכות מורכבות ייצרו נתונים המסודרים באופן טבעי מטריצות גדולות, או באופן כללי במערכים. התמונות שעוברות תהליך זיהוי פנים מסודרות גם הן במטריצות המכילות את הפיקסלים של כל תמונה. תהליך SVD מסייע לנו בחישוב פנים עצמיות הספקות ייצוג יעיל של תמונות הפנים בתהליך זיהוי הפנים. דבר

הדורש כמות גדולה של זיכרון לאחסון התמונה. ע"י כתיבת התמונה בצורתה המפורקת נוכל ליצור מטריצות מממד נמוך יותר [9] (Magaia, M. Herbst), [2] (A. Compton, L. Ernstberger).

בתהליך PCA הרעיון המרכזי הוא לצמצם את הממדיות של מערך הנתונים שיש בו מספר רב של גורמים הקשורים זה בזה. יחד עם זאת, חשוב לנו לשמור על השונות בין משתנים אלו. הפחתה זאת מושגת ע"י הפיכת הנתונים לסט חדש - הרכיבים העיקריים, שאינם מתואמים [5] (T. Jolliffe). שתי שיטות מרכזיות אלו מביאות אותנו לנושא הפנים העצמיות (Eigenfaces). המושג פנים עצמיות מזכיר את המושג וקטורים עצמיים. שיטה זו מבוססת על הפחתת הממד של המרחבים הווקטוריים תוך כדי שמירת עיקרי המידע [9] (Magaia, M. Herbst).

זיהוי פנים

זיהוי הפנים הוא דרך לאשר את זהותו של אדם באמצעות פניו. פניו של אדם מכילות מספר רב של פרטים המבדילים אותן מפניהם של אנשים אחרים. עובדה זו הופכת את הפנים למאפיין ביומטרי היכול לשמש לזיהוי נכון של אדם, לצד מאפיינים ביומטריים אחרים כמו טביעות אצבע, רשתית העין, קול, צורת כף היד, צורת החתימה, ועוד. הצורך במערכות זיהוי פנים נולד בעקבות הצורך לזיהוי ביומטרי ללא ידיעת האדם המזוהה על ידי רשויות אכיפת החוק ברחבי העולם. בהמשך הדרך הפכו אמצעי זיהוי הפנים לאמצעי זיהוי ביומטרי להמונים שלא מצריך את תשומת לב המזוהה אלא מבצע את הזיהוי והגילוי בדרך אגב ולעיתים ללא ידיעת המצולם.

כיום, על מנת לבצע את הזיהוי משתמשים במערכות טכנולוגיות מיוחדות המבוססות מחשב אשר מסוגלות לזהות ולאמת באופן אוטונומי את פניו של אדם ועל ידי כך לאמת את זהותו. ניתן להשתמש במערכות אלו על מנת לזהות אנשים בתמונות, בסרטונים או בזמן אמת. טכנולוגיית זיהוי הפנים משמעותית בחיינו בעיקר לאכיפת חוק ולאבטחה, אך בנוסף בתחומים אחרים שמתפתחים בעולם. בזיהוי פנים נערכת השוואה בין תמונת הפנים של האדם אותו רוצים לזהות ובין מאגר קיים של תמונות פנים. תוצאת הזיהוי היא בדרך כלל של אדם מתוך המאגר שתמונתו היא הדומה ביותר לתמונה שנבדקה.

מערכת כוללת לזיהוי פנים מורכבת משני שלבים עיקריים: גילוי אובייקט הפנים בתוך התמונה הכוללת וזיהוי הפנים.

את שלב הזיהוי ניתן לחלק לשני שלבי ביניים: שלב האימון ושלב הסיווג.

בשלב האימון נעשה שימוש במאגר מוכן של תמונות פנים של אנשים שזהותם ידועה אשר מתוכם יתבצע הזיהוי. מאגר התמונות מכיל תמונה אחת או יותר של כל אדם. בשלב הסיווג נלקחים מאפיינים מהתמונה שנבדקת ונערכת השוואה בינם ובין המאפיינים של כל התמונות האחרות. התוצאה היא התמונה של אדם מתוך המאגר שהמאפיינים שלה הם הדומים ביותר לאלו של התמונה הנבדקת. התוצאה תהיה נכונה כאשר התמונה שנמצאה כדומה ביותר והתמונה שנבדקה הן של אותו אדם.

כיום אנו משתמשים במערכת זיהוי הפנים במקומות רבים: פתיחת הנייד האישי והגנה על נתונים אישיים השמורים בו, כמו כן שמירה על נתונים אישיים במערכות אחרות, אכיפת חוק, שדות תעופה וביקורת גבולות, מציאת נעדרים, צמצום פשיעה קמעונאית, שיפור חוויית הקמעונאות, בנקאות, שיווק ופרסום, בתי חולים המשתמשים במערכת לזיהוי מטופלים, זיהוי רגשות בחולים, מעקב נוכחות, זיהוי נהגים ועוד.

למערכות זיהוי פנים יתרונות רבים ביניהם אבטחה מוגברת גם מבחינה מדינית וגם ברמה האישית, כלי להפחתת פשע, נוחות רבה יותר- מערכת הזיהוי מהירה ונוחה מאשר זיהוי אנושי.

כמו כן, יש גם חסרונות כמו פגיעה בפרטיות, מרחב לטעויות בזיהוי, הצורך באחסון נתונים עצום ועוד.

קיימות מספר שיטות לזיהוי פנים, אשר נבדלות אחת מהשנייה בדרך כלל בשני דברים עיקריים:

- (1) סוג המאפיינים בהן נעשה שימוש ואופן בחירתם מתוך התמונה. השיטה היא טובה יותר ככל שאותם מאפיינים הם ייחודיים עבור כל אדם, ככל שמספרם קטן יותר, וככל שרגישותם לשינויים בתנאי הצילום קטנה יותר.
- (2) בדרך השימוש באותם מאפיינים לשם סיווג התמונה הנבדקת. באופן כללי השיטות הקיימות נחלקות לשיטות המבוססות על מציאת עצמי פנים אופייניים (כמו עיניים, אף, פה, סנטר), לקיחת מרחקים יחסיים ביניהם ושימושם כמאפיינים, ולשיטות סטטיסטיות העושות שימוש בהתפלגות העוצמה על פני התמונה בהן גם נעשתה ההתקדמות הגדולה ביותר במהלך השנים האחרונות שהביאה לשיפור ניכר ביכולת הזיהוי.

רקע מתמטי באלגברה לינארית

1. ערכים עצמיים ווקטורים עצמיים

יהי V מרחב וקטורי ותהי $T: V \rightarrow V$ טרנספורמציה לינארית. אם קיים וקטור $v \in V$ השונה מאפס וסקלר λ כך ש $T(v) = \lambda v$, אזי λ נקרא ערך עצמי של T , ו- v נקרא וקטור עצמי של T השייך לערך עצמי λ .

בהתאם לכך נגדיר ערכים עצמיים ווקטורים עצמיים של מטריצות: תהי A מטריצה ריבועית מסדר n מעל שדה F ויהי $v \in F^n$ וקטור השונה מאפס, אם קיים סקלר λ כך שמתקיים: $Av = \lambda v$ אזי v וקטור עצמי של מטריצה A השייך לערך עצמי λ .

עבור מטריצה A הווקטורים העצמיים המתאימים לערך עצמי λ הם כל הפתרונות של מערכת המשוואות ההומוגנית $(A - \lambda I)v = 0$ כאשר I מטריצה היחידה. למשוואה זאת תהיה פתרון שאינו הטריוויאלי כלומר, $v \neq 0$ רק כאשר $\det(A - \lambda I) = 0$. אוסף הפתרונות נקרא המרחב העצמי של λ והוא תמיד מרחב וקטורי. אם קיימים פתרונות לא טריוויאליים אזי λ הוא ערך עצמי וממדו של המרחב נקרא הריבוי הגיאומטרי של הסקלר.

וקטורים עצמיים של ערכים עצמיים שונים הם בלתי תלויים לינארית זה בזה.

2. מכפלה סקלרית

יהיו שני וקטורים $\vec{x}$ ו- $\vec{y}$, כאשר: $\vec{x} = [x_1 \ \dots \ x_n]^T$, $\vec{y} = [y_1 \ \dots \ y_n]^T$. המכפלה הסקלרית בין שני הוקטורים:

$$\vec{x} \cdot \vec{y} = \vec{x}^T \vec{y} = x_1 y_1 + x_2 y_2 + \dots + x_n y_n.$$

3. נורמה של וקטור

יהי $\vec{x} \in \mathbb{R}^n$, הנורמה האוקלידית של וקטור $\vec{x}$ מוגדרת בצורה:

$$(1) \quad \|\vec{x}\|_2 = (\vec{x}^T \vec{x})^{1/2} = (\vec{x}_1^2 + \vec{x}_2^2 + \dots + \vec{x}_n^2)^{1/2}$$

4. אורתוגונליות ואורתונורמליות

אורתוגונליות בין וקטורים:

יהיו שני וקטורים, וקטור $\vec{u}$ ווקטור $\vec{v}$. נאמר כי הוקטורים אורתוגונליים זה לזה אם מתקיים:

$$\vec{u} \perp \vec{v} \Leftrightarrow \vec{u} \cdot \vec{v} = 0$$

אורתונורמליות בין וקטורים:

תהי $\{\vec{x}_1, \dots, \vec{x}_n\}$ קבוצת וקטורים ב- $\mathbb{R}^n$. נאמר כי הקבוצה אורתונורמלית אם מתקיים:

$$\vec{x}_i \cdot \vec{x}_j = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$$

מטריצה אורתוגונלית:

מטריצה אורתוגונלית היא מטריצה ריבועית שרכיביה ממשיים, המקיימת את התנאי: $A^T A = A A^T = I$ כאשר I היא מטריצת היחידה, ו- A^T היא המטריצה המשוחלפת של A . עמודות המטריצה האורתוגונלית מהוות בסיס אורתונורמלי למרחב הוקטורי שממנו כמספר עמודות המטריצה, עם המכפלה הפנימית הסטנדרטית.

5. נורמה של מטריצה

תהי A מטריצה מסדר $m \times n$, קיימים סוגים שונים של נורמות של מטריצה, נתמקד בשניים שרלוונטיים עבורנו:

- נורמת L_2 :

$$(2) \quad \|A\|_2 = \sup_{\vec{x} \neq 0} \frac{\|A\vec{x}\|_2}{\|\vec{x}\|_2} = \sup_{\{\|\vec{x}\|_2=1\}} \|A\vec{x}\|_2$$

- נורמת פרובניוס:

$$(3) \quad \|A\|_F = \left(\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2 \right)^{1/2} = \sqrt{\text{tr}(A^T A)} = \sqrt{\text{tr}(A A^T)}$$

כאשר $\text{tr} = \text{trace}$ – עקבה של מטריצה, זאת אומרת סכום האיברים באלכסון הראשי של המטריצה.

תכונות הקשורות לנורמה של מטריצה:**טענה 1:**

יהיו A ו- B מטריצות, כך ש AB מוגדר, ותהי Q מטריצה אורתוגונלית. התכונות הבאות מתקיימות:

- א. $\|AB\|_2 \leq \|A\|_2 \|B\|_2$.
- ב. $\|Q\vec{x}\|_2 = \|\vec{x}\|_2$ לכל $\vec{x}$.
- ג. $\|QA\|_2 = \|A\|_2$.
- ד. $\|AQ\|_2 = \|A\|_2$.
- ה. $\|QA\|_F = \|AQ\|_F = \|A\|_F$.

הוכחה :

א. מכיוון שלכל $\vec{x}$ מתקיים $\|A\vec{x}\|_2 \leq \|A\|_2 \|\vec{x}\|_2$, אזי

$$\sup_{\|\vec{x}\|=1} \|A\vec{x}\|_2 = \sup_{\|\vec{x}\|=1} \|A(B\vec{x})\|_2 \leq \|A\|_2 \sup_{\|\vec{x}\|=1} \|B\vec{x}\|_2 = \|A\|_2 \|B\|_2$$

$$\|Q\vec{x}\|_2 = (Q\vec{x}^T \cdot Q\vec{x})^{1/2} = (x^T Q^T Q x)^{1/2} = (x^T x)^{1/2} = \|\vec{x}\|_2 \quad \text{ב.}$$

ג. נשתמש בסעיף ב',

$$\|QA\|_2 = \sup_{\|\vec{x}\|=1} \|Q(A\vec{x})\|_2 = \sup_{\|\vec{x}\|=1} \|(A\vec{x})\|_2 = \|A\|_2$$

ד. מכיוון שלכל $\vec{x}$ מתקיים $\|\vec{x}\|_2 = \|Q\vec{x}\|_2$, אז

$$\sup_{\|\vec{x}\|=1} \|AQ\vec{x}\|_2 = \sup_{\|Q\vec{x}\|=1} \|A(Q\vec{x})\|_2 = \|A\|_2$$

$$\begin{aligned} \|QA\|_F^2 &= \text{tr}((QA)^T QA) = \text{tr}(A^T Q^T QA) = \text{tr}(A^T A) = \|A\|_F^2 \\ \|AQ\|_F^2 &= \text{tr}((AQ)^T AQ) = \text{tr}(A^T Q^T QA) = \text{tr}(A^T A) = \|A\|_F^2 \end{aligned} \quad \text{ה.}$$

פירוק SVD

פירוק SVD – Singular value decomposition שפירושו הוא פירוק לערכים סינגולריים, זוהי שיטת פירוק באלגברה לינארית של מטריצה מרוכבת או ממשית. לשיטה זו שימושים רבים בעיבוד אותות וסטטיסטיקה. בעבודה זו נעשה שימוש בפירוק SVD כשיטה להורדת ממד של מטריצה. פירוק זה הוא אלגוריתם מרכזי בתיאוריית PCA, שבה נתונים בממד גבוה מפורקים לגורמים תיאוריים מבחינה סטטיסטית. PCA ו-SVD חלים במגוון רחב של בעיות בעולם המדע וההנדסה.

רקע היסטורי

ל-SVD יש היסטוריה ארוכה ועשירה, החל מעבודה מוקדמת לפיתוח התיאורטי של היסודות ועד לעבודה מודרנית על יציבות ויעילות חישובית. תחילה עשו את עבודתם התיאורטית המוקדמת בתחום בלטרמי וג'ורדן (1873), סילבסטר (1889), שמידט (1907), ווייל (1912). עבודה עדכנית יותר, כולל העבודה החישובית המכוננת נעשתה ע"י גולוב ומשתפי הפעולה.

הגדרה פורמלית

בהינתן מטריצה $A \in \mathbb{R}^{m \times n}$ בהנחה כי m ו- n שלמים חיוביים, פירוק ה-SVD של מטריצה A הוא המכפלה:

$$(4) \quad A = U \Sigma V^T$$

כאשר:

- $U \in \mathbb{R}^{m \times m}$ מטריצה אורתוגונלית.
- $V \in \mathbb{R}^{n \times n}$ מטריצה אורתוגונלית.
- $\Sigma \in \mathbb{R}^{m \times n}$ מטריצה "כמעט אלכסונית", כאשר $\Sigma_{ij} = \begin{cases} \sigma_i, & i = j \\ 0, & i \neq j \end{cases}$

איברי האלכסון של מטריצה Σ הם מספרים אי-שליליים המסומנים ב- σ_i ונקראים הערכים הסינגולריים של מטריצה A , מסודרים בסדר יורד $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0$ כאשר $p = \min(m, n)$. אם נמין אותם הערכים השווים לאפס כולם נמצאים בעמודות או בשורות הממוספרות גבוה (זאת אומרת בסוף). העמודות של מטריצה U והעמודות של מטריצה V יוצרות בהתאמה שתי קבוצות של בסיסים אורתונורמליים $\{\vec{u}_1, \dots, \vec{u}_m\}$ של $\mathbb{R}^m$ ו- $\{\vec{v}_1, \dots, \vec{v}_n\}$ של $\mathbb{R}^n$ ושל $\mathbb{R}^n$.

נוכל לרשום את פירוק ה-SVD בצורה הבאה:

$$(5) \quad A = \sum_{i=1}^p \sigma_i \vec{u}_i \vec{v}_i^T$$

פירוק SVD מנקודת מבט גיאומטרית

כפי שאמרנו, העמודות של מטריצה U ושל מטריצה V יוצרות בהתאמה שתי קבוצות של בסיסים אורתונורמליים $\{\vec{u}_1, \dots, \vec{u}_m\}$ של $\mathbb{R}^m$ ו- $\{\vec{v}_1, \dots, \vec{v}_n\}$ של $\mathbb{R}^n$.

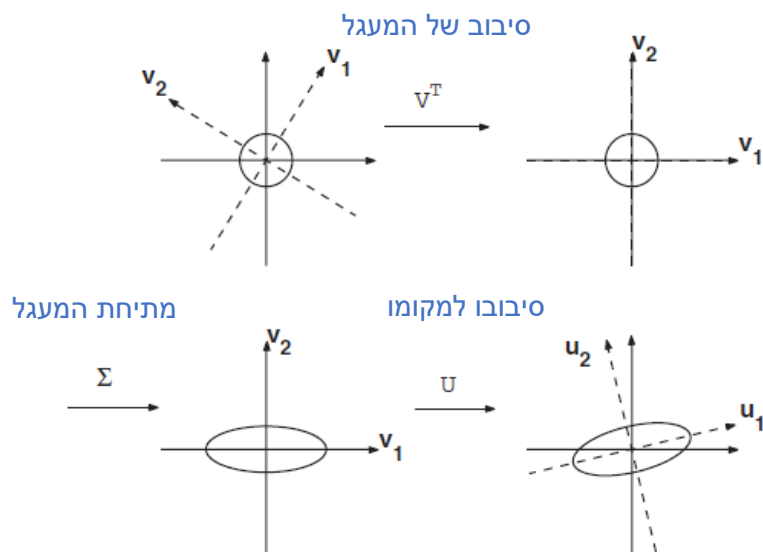
באמצעות פירוק SVD ניתן לתאר בקלות את התמונה של ספירת היחידה תחת ההעתקה $A: \mathbb{R}^n \rightarrow \mathbb{R}^m$. ספירת היחידה היא $S^{n-1} = \{\vec{x} \in \mathbb{R}^n: \|\vec{x}\|_2 = 1\}$. מהתיאור למעלה נקבל כי AS^{n-1} זה אליפסואיד עם צירים ראשיים בכיוון $\vec{u}_i$, כאשר אורכי הצירים הראשיים σ_i .

בהנחה ש- $\sigma_p > 0$,

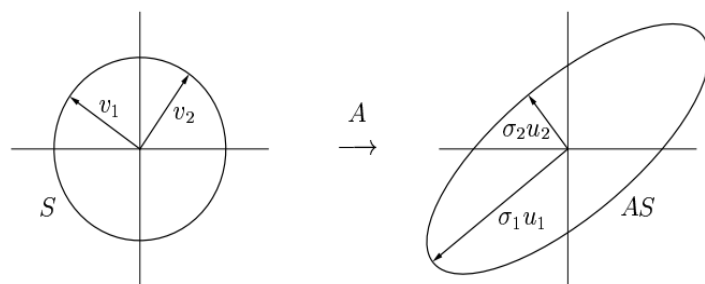
$$(6) \quad AS^{n-1} = \left\{ \frac{w_1^2}{\sigma_1^2} + \dots + \frac{w_p^2}{\sigma_p^2} \leq 1 \right\}$$

כאשר $\{w_1, \dots, w_p\}$ הן הקואורדינטות בבסיס אורתונורמלי $\{\vec{u}_1, \dots, \vec{u}_p\}$.

המטריצה האורתוגונלית V^T גורמת לשימור צורת הכדור, המטריצה האלכסונית Σ גורמת למתיחה עד הפיכת הצורה לאליפסואיד המיושרת עם הבסיס הקנוני, והמטריצה האורתוגונלית U גורמת לסיבוב האליפסואיד ללא שינוי הצורה.



לכן, אם נוכל להוכיח שלכל מטריצה יש פירוק SVD, נוכל להוכיח כי הפעלה לינארית של מטריצה A על ספירת היחידה גורמת לקבלת תמונה של אליפסואיד.



פירוק SVD של מטריצה 2×2

פירוק SVD מצומצם

ראשית נדון בקשר בין הפירוק לדרגה של המטריצה A . הפירוק $A = U\Sigma V^T$ שקול לכך ש:

$$(7) \quad AV = U\Sigma = [\sigma_1 \vec{u}_1 \dots \sigma_p \vec{u}_p]$$

כאשר $p = \min(m, n)$ ו- $\sigma_p \geq 0$ ו- $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p$.

מכיוון ש- $rank(V) = n$ מהמשוואה (7) נובע כי:

$$rank(A) = rank(AV) = rank(U\Sigma) = \{\sigma_i \neq 0 \text{ מספר}\}$$

נציב $r = rank(A)$, אזי $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$ ו- $\sigma_i = 0$ עבור $i = r + 1, \dots, p$.

לפיכך, $AV = [\sigma_1 \vec{u}_1 \dots \sigma_p \vec{u}_r \underbrace{0 \dots 0}_{p-r}]$ ומכאן נקבל כי,

$$A = AVV^T = (\sigma_1 \vec{u}_1 \dots \sigma_p \vec{u}_r, 0, \dots, 0)V^T = \sigma_1 \vec{u}_1 \vec{v}_1^T + \dots + \sigma_r \vec{u}_r \vec{v}_r^T$$

קיבלנו צורה מצומצמת לפירוק SVD :

$$(8) \quad A = \sum_{i=1}^r \sigma_i \vec{u}_i \vec{v}_i^T,$$

כאשר $r = \text{Rank}(A)$.

לכן ניתן לראות כי, פירוק SVD מצומצם מתקבל מפירוק SVD המלא באופן הבא :

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \ddots & 0 & \vdots \\ 0 & 0 & \sigma_r & \vdots \\ 0 & \dots & \dots & 0 \end{bmatrix}$$

$$A = U\Sigma V^T = [\vec{u}_1 \dots \vec{u}_m] \begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \ddots & 0 & \vdots \\ 0 & 0 & \sigma_r & \vdots \\ 0 & \dots & \dots & 0 \end{bmatrix} \begin{bmatrix} \vec{v}_1^T \\ \vdots \\ \vec{v}_n^T \end{bmatrix} = [\vec{u}_1 \dots \vec{u}_r \quad \vec{u}_{r+1} \dots \vec{u}_m] \begin{bmatrix} \sigma_1 \vec{v}_1^T \\ \vdots \\ \sigma_r \vec{v}_r^T \\ 0 \\ \vdots \\ 0 \end{bmatrix} =$$

$$= \sigma_1 \vec{u}_1 \vec{v}_1^T + \dots + \sigma_r \vec{u}_r \vec{v}_r^T.$$

משפט הקיום

לכל מטריצה $A \in \mathbb{R}^{m \times n}$ קיים פירוק SVD.

הוכחת משפט הקיום

כדי להוכיח את הקיום של פירוק SVD נגדיר את הערך הסינגולרי הגדול ביותר של מטריצה A על ידי $\|A\|_2 = \sigma_1$,

$$\sigma_1 = \sup_{\|\vec{x}\|_2=1} \|A\vec{x}\|_2.$$

כעת נרצה להראות כי קיים וקטור $\vec{v}_1 \in \mathbb{R}^n$ כך ש- $\|\vec{v}_1\| = 1$ ו- $\|A\vec{v}_1\|_2 = \sigma_1$, לשם כך ניקח סדרת וקטורים $\{\vec{v}_n\}_{n=2}^\infty$ כך ש $\|\vec{v}_n\|_2 = 1$ ו- $\|A\vec{v}_n\|_2 = \sigma_1$.

מכיוון ש- $\{\|\vec{x}\|_2 = 1\}$ קבוצה סגורה וחסומה, אזי קיימת תת סדרה מתכנסת: $\lim_{k \rightarrow \infty} \vec{v}_{n_k} = \vec{v}$,

וזהו הוקטור $\vec{v}_1$. כעת נציב: $\vec{u}_1 = \frac{1}{\sigma_1} A\vec{v}_1$ אז $\|\vec{u}_1\|_2 = \frac{1}{\sigma_1} \|A\vec{v}_1\|_2$.

באמצעות תהליך גרם שמידט נבחר בסיסים אורתוגונליים ל- $\mathbb{R}^n$ ו- $\mathbb{R}^m$ באופן הבא:

$\{\vec{v}_1, \vec{x}_2, \dots, \vec{x}_n\}$ בסיס אורתונורמלי ל- $\mathbb{R}^n$ ו- $\{\vec{u}_1, \vec{y}_2, \dots, \vec{y}_m\}$ ל- $\mathbb{R}^m$. נגדיר בהתאמה מטריצות אורתוגונליות $U_1 = [\vec{u}_1 \ \vec{y}_2 \ \dots \ \vec{y}_m]$, $V_1 = [\vec{v}_1 \ \vec{x}_2 \ \dots \ \vec{x}_n]$.

כעת נחשב $U_1^T A V_1$. מכיוון ש $A \vec{v}_1 = \sigma_1 \vec{u}_1$ ו- $\vec{y}_i^T \cdot \vec{u}_1 = 0$ אזי

$$U_1^T A V_1 = \begin{bmatrix} \vec{u}_1^T \\ \vec{y}_2^T \\ \vdots \\ \vec{y}_m^T \end{bmatrix} [A \vec{v}_1 \quad A \vec{x}_2 \quad \dots \quad A \vec{x}_n] = \begin{bmatrix} \vec{u}_1^T \\ \vec{y}_2^T \\ \vdots \\ \vec{y}_m^T \end{bmatrix} \begin{bmatrix} \sigma_1 \vec{u}_1 & A \vec{x}_2 & \dots & A \vec{x}_n \end{bmatrix}$$

$$= \begin{bmatrix} \sigma_1 \vec{u}_1^T \vec{u}_1 & \vec{u}_1^T A \vec{x}_2 & \dots & \vec{u}_1^T A \vec{x}_n \\ \vec{y}_2^T \sigma_1 \vec{u}_1 & & & \\ \vdots & & B & \\ \vec{y}_m^T \sigma_1 \vec{u}_1 & & & \end{bmatrix} = \begin{bmatrix} \sigma_1 & \overbrace{\vec{u}_1^T A \vec{x}_2 \quad \dots \quad \vec{u}_1^T A \vec{x}_n}^{\vec{\omega}^T} \\ 0 & \\ \vdots & \\ 0 & \end{bmatrix} = \begin{bmatrix} \sigma_1 & \vec{\omega}^T \\ 0 & B \end{bmatrix}.$$

נציב $S = \begin{bmatrix} \sigma_1 & \vec{\omega}^T \\ 0 & B \end{bmatrix}$, אז לפי החישוב מעלה

$$U_1^T A V_1 = S,$$

כאשר מטריצה B מסדר $(n-1) \times (m-1)$. כעת נראה ש $\vec{\omega} = 0$. נפעיל את S על וקטור $\begin{bmatrix} \sigma_1 \\ \vec{\omega} \end{bmatrix}$:

$$S \begin{bmatrix} \sigma_1 \\ \vec{\omega} \end{bmatrix} = \begin{bmatrix} \sigma_1 & \vec{\omega}^T \\ 0 & B \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \vec{\omega} \end{bmatrix} = \begin{bmatrix} \sigma_1^2 + \|\vec{\omega}\|_2^2 \\ B \vec{\omega} \end{bmatrix},$$

אז,

$$\left\| S \begin{bmatrix} \sigma_1 \\ \vec{\omega} \end{bmatrix} \right\|_2^2 = (\sigma_1^2 + \|\vec{\omega}\|_2^2)^2 + \|B \vec{\omega}\|_2^2 \geq (\sigma_1^2 + \|\vec{\omega}\|_2^2)^2$$

ולכן,

$$\frac{\left\| S \begin{bmatrix} \sigma_1 \\ \vec{\omega} \end{bmatrix} \right\|_2}{(\sigma_1^2 + \|\vec{\omega}\|_2^2)^{1/2}} \geq (\sigma_1^2 + \|\vec{\omega}\|_2^2)^{1/2}$$

לפי הגדרת הנורמה (2):

$$\|S\|_2 \geq (\sigma_1^2 + \|\vec{\omega}\|_2^2)^{1/2}$$

אבל מאחר ו- U_1 וגם V_1 אורתוגונליות, אזי לפי תכונות הנורמה (טענה 1):

$$\sigma_1 = \|A\|_2 = \|U_1^T A V_1\|_2 = \|S\|_2 \geq (\sigma_1^2 + \|\vec{\omega}\|_2^2)^{1/2}$$

ואי שוויון זה יכול להתקיים אם ורק אם $\vec{\omega} = 0$. לפיכך, $S = \begin{bmatrix} \sigma_1 & 0 \\ 0 & B \end{bmatrix}$.

מכיוון שמטריצה B מסדר $(m-1) \times (n-1)$, אז לפי הנחת האינדוקציה יש לה פירוק SVD:

$$B = U_2 \Sigma_2 V_2^T.$$

לפיכך,

$$A = U_1 S V_1^T = U_1 \begin{bmatrix} 1 & 0 \\ 0 & U_2 \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & V_2 \end{bmatrix}^T V_1^T$$

נשים לב כי מטריצה U_2 אורתוגונלית מסדר $(m-1) \times (m-1)$, לכן $\begin{bmatrix} 1 & 0 \\ 0 & U_2 \end{bmatrix}$ מטריצה אורתוגונלית מסדר $m \times m$. מכיוון שכפל של מטריצות אורתוגונליות הוא מטריצה אורתוגונלית. כנ"ל עבור $\begin{bmatrix} 1 & 0 \\ 0 & V_2 \end{bmatrix}$ אורתוגונלית מסדר $n \times n$. נותר להראות את השלב הראשוני של האינדוקציה:

אם $n=1$ אז $A = \vec{a} \in \mathbb{R}^m$, נציב $\sigma_1 = \|\vec{a}\|_2$, $U = \frac{1}{\sigma_1}$ ו- $V = [1]$, אזי

$$A = \vec{a} = \sigma_1 U V^T.$$

באופן דומה עבור $m=1$, ובכך נסיים את הוכחת האינדוקציה עבור קיום של פירוק SVD.

חישוב נורמות ע"י פירוק SVD

נורמת L_2 :

לפי הוכחת משפט הקיום,

$$(9) \quad \|A\|_2 = \sigma_1,$$

הערך הסינגולרי הגדול ביותר.

נורמת פרובניוס:

$$(10) \quad \|A\|_F = \|U \Sigma V^T\|_F = \|\Sigma\|_F = (\sigma_1^2 + \sigma_2^2 + \dots + \sigma_r^2)^{1/2}.$$

דוגמא לפירוק SVD של מטריצה A ממשית

$$A = \begin{bmatrix} 5 & 5 \\ -1 & 7 \end{bmatrix}.$$

נשים לב כי לפי פירוק SVD (4),

$$A^T A = (V \Sigma U^T)(U \Sigma V^T) = V(\Sigma^T \Sigma) V^T,$$

כאן $\Sigma^T \Sigma$ מטריצה $n \times n$ אלכסונית עם σ_j^2 איברי האלכסון. לכן הוקטורים $\{\vec{v}_1, \dots, \vec{v}_n\}$ הם וקטורים עצמיים של מטריצה $A^T A$.

$$A^T A = \begin{bmatrix} 5 & -1 \\ 5 & 7 \end{bmatrix} \begin{bmatrix} 5 & 5 \\ -1 & 7 \end{bmatrix} = \begin{bmatrix} 26 & 18 \\ 18 & 74 \end{bmatrix}$$

נחשב את הפולינום האופייני:

$$\det(A^T A - \lambda I) = \det \begin{pmatrix} 26 - \lambda & 18 \\ 18 & 74 - \lambda \end{pmatrix} = \lambda^2 - 100\lambda + 1600 = (\lambda - 80)(\lambda - 20)$$

$$\sigma_1^2 = 80 \text{ ו- } \sigma_2^2 = 20.$$

נחשב את הוקטורים העצמיים,

$$\begin{bmatrix} 26 - 80 & 18 \\ 18 & 74 - 80 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -54 & 18 \\ 18 & -6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

$$\vec{v}_1 = \frac{1}{\sqrt{10}} \begin{bmatrix} 1 \\ 3 \end{bmatrix} \text{ ולאחר נרמול } \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \text{ אז,}$$

$$\vec{v}_2 = \frac{1}{\sqrt{10}} \begin{bmatrix} -3 \\ 1 \end{bmatrix} \text{ לכן } \vec{v}_1 - \vec{v}_2$$

$$V = \begin{bmatrix} 1 & -3 \\ 3 & 1 \end{bmatrix}.$$

המטריצה

כעת,

$$\vec{u}_1 = \frac{1}{\sigma_1} A \vec{v}_1 = \frac{1}{80} \begin{bmatrix} 5 & 5 \\ -1 & 7 \end{bmatrix} \frac{1}{\sqrt{10}} \begin{bmatrix} 1 \\ 3 \end{bmatrix} = \frac{1}{20\sqrt{2}} \begin{bmatrix} 20 \\ 20 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

$$\vec{u}_2 = \frac{1}{\sigma_2} A \vec{v}_2 = \frac{1}{\sqrt{20}} \begin{bmatrix} 5 & 5 \\ -1 & 7 \end{bmatrix} \frac{1}{\sqrt{10}} \begin{bmatrix} -3 \\ 1 \end{bmatrix} = \frac{1}{10\sqrt{2}} \begin{bmatrix} -10 \\ 10 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 1 \end{bmatrix}.$$

לסיכום,

$$U = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}, V = \frac{1}{\sqrt{10}} \begin{bmatrix} 1 & -3 \\ 3 & 1 \end{bmatrix}$$

-1

$$\Sigma = \begin{bmatrix} \sqrt{80} & 0 \\ 0 & \sqrt{20} \end{bmatrix}$$

Principal Component Analysis

PCA - principal component analysis, בשפה העברית ניתוח רכיבים העיקריים, היא שיטה להתמרה לינארית של נתונים למערכת קואורדינטות חדשה שבה המידע בקואורדינטות החדשות הוא אורתוגונלי ובעל שונות הולכת וקטנה.

המטרה העיקרית בשימוש בשיטת PCA היא הפחתת הממד של מערך הנתונים הנתון בו קשורים מספר רב של משתנים, תוך שמירה על עיקרי הנתונים הנמצאים בו.

בשיטה זו המטרה מושגת ע"י הפיכתם של המשתנים לקבוצה חדשה של משתנים (הרכיבים העיקריים) ללא מתאם משותף, ומסודרים כך שהחלק הראשון במערך הנתונים החדש ישמור על רוב השונות שהיו קיימות בין המשתנים המקוריים.

PCA הוא גם אחד מהשימושים המרכזיים של SVD, סידור זה של מערך הנתונים כולל את מטריצת השונות המשותפת. תהליך PCA מעבד מראש את הנתונים ע"י חיסור ממוצע לפני ביצוע פירוק SVD. הגיאומטריה של מערכת הקואורדינטות נקבעת ע"י הרכיבים העיקריים אשר אינם מתואמים (אורתוגונליים) זה לזה, אך בעלי מתאם מקסימלי עם המדידות.

תיאוריה זו פותחה בשנת 1901 ע"י פירסון, ובאופן עצמאי ע"י הוטלינג בשנות ה-30.

ממוצע ומטריצת השונות המשותפת



הפיזור A של וקטורי התצפית $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n$.

כדי להתכונן לתהליך מציאת הרכיבים העיקריים (PCA), ניקח את $[\vec{x}_1 \ \vec{x}_2 \ \dots \ \vec{x}_n]$ להיות מטריצה מסדר $n \times p$, כאשר n מספר התצפיות ו- p מספר המדדים.

ממוצע המדגם $\vec{M}$, של וקטורי התצפית $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n$ מיוצגת באופן הבא:

$$(11) \quad \vec{M} = \frac{1}{n} (\vec{x}_1 + \vec{x}_2 + \dots + \vec{x}_n)$$

עבור הנתונים בתמונה לעיל, ממוצע המדגם יהיה הנקודה המרכזית בגרף הפיזור שאנו רואים.

עבור $k = 1, \dots, n$ נקבל:

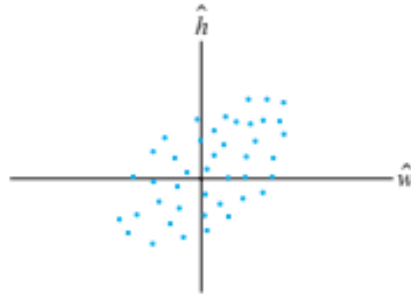
$$(12) \quad \hat{x}_k = \vec{x}_k - \vec{M}$$

ו- $n \times p$ וקטורי העמודות יהיו מיוצגות במטריצה :

$$B = [\hat{x}_1 \quad \hat{x}_2 \quad \dots \quad \hat{x}_n],$$

כאשר ממוצע המדגם יהיה שווה לאפס, B מייצגת תבנית של סטייה ממוצעת.

כאשר הורדנו את ממוצע הפיזור מהמדגם, נקבל פיזור כמו באיור הבא :



הפיזור B של וקטורי התצפית $\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n$ כאשר הממוצע הוא 0.

מטריצת השונות המשותפת

מטריצת השונות המשותפת (מדגם) מוגדרת ע"י :

$$(13) \quad S = \frac{1}{n} B B^T = \frac{1}{n} \sum_{k=1}^n \hat{x}_k \hat{x}_k^T$$

לפי הגדרת המינוח באנגלית Covariance matrix.

מאחר וכל מטריצה מהצורה $B B^T$ היא סימטרית ומוגדרת חיובית למחצה, אז S מטריצה סימטרית ומוגדרת חיובית למחצה גם כן.

נציין כי את מטריצת השונות המשותפת S ניתן להגדיר בשני אופנים :

$$(14) \quad S = \frac{1}{n} \sum_{j=1}^n \vec{x}_j \vec{x}_j^T$$

$$(15) \quad S = \frac{1}{n-1} \sum_{j=1}^n \vec{x}_j \vec{x}_j^T$$

בעבודה זו נעבוד עם ייצוג (14) עבור מטריצת השונות המשותפת.

דוגמא למציאת מטריצת שונות משותפת

נניח כי יש לנו $p=3$ תצפיות שנעשו על $n=4$ פריטים:

$$\vec{x}_1 = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}, \quad \vec{x}_2 = \begin{bmatrix} 4 \\ 2 \\ 13 \end{bmatrix}, \quad \vec{x}_3 = \begin{bmatrix} 7 \\ 8 \\ 1 \end{bmatrix}, \quad \vec{x}_4 = \begin{bmatrix} 8 \\ 4 \\ 5 \end{bmatrix}$$

ממוצע המדגם:

$$\vec{M} = \frac{1}{4} \left(\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} + \begin{bmatrix} 4 \\ 2 \\ 13 \end{bmatrix} + \begin{bmatrix} 7 \\ 8 \\ 1 \end{bmatrix} + \begin{bmatrix} 8 \\ 4 \\ 5 \end{bmatrix} \right) = \frac{1}{4} \begin{bmatrix} 20 \\ 16 \\ 20 \end{bmatrix} = \begin{bmatrix} 5 \\ 4 \\ 5 \end{bmatrix}$$

נפחית את ממוצע המדגם מכל התצפיות:

$$\hat{x}_1 = \begin{bmatrix} 1-5 \\ 2-4 \\ 1-5 \end{bmatrix} = \begin{bmatrix} -4 \\ -2 \\ -4 \end{bmatrix}, \quad \hat{x}_2 = \begin{bmatrix} 4-5 \\ 2-4 \\ 13-5 \end{bmatrix} = \begin{bmatrix} -1 \\ -2 \\ 8 \end{bmatrix},$$

$$\hat{x}_3 = \begin{bmatrix} 7-5 \\ 8-4 \\ 1-5 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ -4 \end{bmatrix}, \quad \hat{x}_4 = \begin{bmatrix} 8-5 \\ 4-4 \\ 5-5 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \\ 0 \end{bmatrix}$$

ולכן נקבל:

$$B = \begin{bmatrix} -4 & -1 & 2 & 3 \\ -2 & -2 & 4 & 0 \\ -4 & 8 & -4 & 0 \end{bmatrix}$$

מטריצת השונות המשותפת (של המדגם) שמתקבלת:

$$\begin{aligned} S &= \frac{1}{4} \begin{bmatrix} -4 & -1 & 2 & 3 \\ -2 & -2 & 4 & 0 \\ -4 & 8 & -4 & 0 \end{bmatrix} \begin{bmatrix} -4 & -2 & -4 \\ -1 & -2 & 8 \\ 2 & 4 & -4 \\ 3 & 0 & 0 \end{bmatrix} = \\ &= \frac{1}{4} \begin{bmatrix} 30 & 18 & 0 \\ 18 & 24 & -24 \\ 0 & -24 & 96 \end{bmatrix} = \begin{bmatrix} \frac{15}{2} & \frac{9}{2} & 0 \\ 9 & 6 & -6 \\ 0 & -6 & 24 \end{bmatrix} \end{aligned}$$

לצורך לדוגמא כללית ניקח:

$$\vec{x}_1 = \begin{bmatrix} x_{11} \\ x_{21} \end{bmatrix}, \vec{x}_2 = \begin{bmatrix} x_{12} \\ x_{22} \end{bmatrix}, \vec{x}_3 = \begin{bmatrix} x_{13} \\ x_{23} \end{bmatrix}$$

ונניח כי הממוצע הוא אפס.

אזי,

$$S = \frac{1}{3} \begin{bmatrix} \sum_{k=1}^3 x_{1k}^2 & \sum_{k=1}^3 x_{1k}x_{2k} \\ \sum_{k=1}^3 x_{2k}x_{1k} & \sum_{k=1}^3 x_{2k}^2 \end{bmatrix}$$

נדון במשמעות המקדים של מטריצה S:

s_{11} תייצג את תייצג את השונות של הדגימה $(x_{11}, \dots, x_{1n})$.

s_{12} היא השונות המשותפת בין (x_{11}, x_{12}, x_{13}) ל- (x_{21}, x_{22}, x_{23}) .

אם למשל $s_{12} = 0$, אז לא קיים מתאם בין שני המדדים, ניתן לאמר כי שני המדדים אורתוגונליים.

באופן כללי כאשר מטריצה S מסדר $p \times p$, אזי s_{jj} השונות של מדד j. s_{ij} המתאם של מדד i וממד j (עבור $i \neq j$).

השונות הכוללת של הנתונים היא סכום השונות על האלכסון הראשי של מטריצה S. ידוע כי סכום האיברים באלכסון הראשי של מטריצה S נקרא העקבה של המטריצה ונכתב $\text{tr}(S)$ (trace).

$$\{ \text{השונות הכוללת} \} = \text{tr}(S)$$

תהליך PCA

לשם הפשטות, נניח כי הממוצע של העמודות של המטריצה $[\vec{x}_1 \quad \vec{x}_2 \quad \dots \quad \vec{x}_n]$ הוא אפס.

המטרה של PCA היא למצוא מטריצה אורתוגונלית מסדר $p \times p$, $U = [\vec{u}_1 \quad \vec{u}_2 \quad \dots \quad \vec{u}_p]$, שקובעת את השינוי של משתנה $\vec{x} = U\vec{y}$ כלומר,

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix} = [\vec{u}_1 \quad \vec{u}_2 \quad \dots \quad \vec{u}_p] \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{bmatrix}.$$

כאשר המשתנים החדשים $\vec{y}_1, \vec{y}_2, \dots, \vec{y}_p$ לא מתואמים ומסודרים בסדר יורד של השונות.

השינוי האורתוגונלי של המשתנה $\vec{x}_k = U\vec{y}_k$ אומר שכל וקטור תצפית $\vec{x}_n$ מקבל "שם חדש" $\vec{y}_k$

נחשב את מטריצת השונות המשותפת המשתנים $\vec{y}_1, \vec{y}_2, \dots, \vec{y}_k$:

$$Y = U^T X$$

$$YY^T = U^T X (U^T X)^T = U^T X X^T U$$

קיבלנו, $S_Y = U^T S_X U$.

$$S_X = \frac{1}{n} \sum_{k=1}^n \vec{x}_k \vec{x}_k^T = \frac{1}{n} \sum_{k=1}^n (U \vec{y}_k) (U \vec{y}_k)^T = \frac{1}{n} \sum_{k=1}^n U \vec{y}_k \vec{y}_k^T U^T = U S_Y U^T$$

אז מטריצה U שאנו מחפשים היא מטריצה שמלכסנת את מטריצה השונות המשותפת $U^T S_X U$.

קיימת מטריצה U אורתוגונלית כך ש- $U^T S_X U = D$, כאשר D מטריצה אלכסונית.

מכיוון שמטריצה S_X היא מטריצה סימטרית, אז על סמך המשפט של פירוק ספקטרלי, U תמיד קיימת והעמודות שלה הם הוקטורים העצמיים של מטריצה S_X .

נקבל כי :

$$(16) \quad S_X = U D U^T$$

$$(17) \quad D = U^T S_X U = S_Y$$

דוגמא למציאת רכיבים עיקריים

נניח כי קיימת מטריצה S , שהיא מטריצת השונות המשותפת של מדגם וקטורים במרחב $\mathbb{R}^3$:

$$S = \begin{bmatrix} 2382.78 & 2611.84 & 2136.20 \\ 2611.84 & 3106.47 & 2553.90 \\ 2136.20 & 2553.90 & 2650.71 \end{bmatrix}$$

נמצא את הערכים העצמיים של מטריצה S :

$$S - \lambda I = \begin{bmatrix} 2382.78 - \lambda & 2611.84 & 2136.20 \\ 2611.84 & 3106.47 - \lambda & 2553.90 \\ 2136.20 & 2553.90 & 2650.71 - \lambda \end{bmatrix}$$

$$\det(S - \lambda I) = 0$$

הפולינום האופייני שנקבל :

$$-\lambda^3 + 8139.96\lambda^2 - 4.04498 \times 10^6 \lambda + 3.19433 \times 10^8 = 0$$

$$\sigma_1^2 = 7614.23, \quad \sigma_2^2 = 427.625, \quad \sigma_3^2 = 98.1048$$

נמצא את הוקטורים העצמיים המנורמלים בהתאמה לערכים העצמיים שמצאנו :

$$u_1 = \begin{bmatrix} 0.541 \\ 0.629 \\ 0.557 \end{bmatrix}, \quad u_2 = \begin{bmatrix} -0.489 \\ -0.302 \\ 0.817 \end{bmatrix}, \quad u_3 = \begin{bmatrix} 0.683 \\ -0.715 \\ 0.144 \end{bmatrix}.$$

כאשר u_1, u_2, u_3 הם הרכיבים העיקריים (PC):

$$U = \begin{bmatrix} 0.541 & -0.489 & 0.683 \\ 0.629 & -0.302 & -0.715 \\ 0.557 & 0.817 & 0.144 \end{bmatrix}$$

נוכל לחשב את מטריצה D , שהיא בעצם מטריצת השונות המשותפת של הנתונים בבסיס אורתונורמלי $\{\vec{u}_1, \vec{u}_2, \vec{u}_3\}$:

$$D = \begin{bmatrix} 7614.23 & 0 & 0 \\ 0 & 427.625 & 0 \\ 0 & 0 & 98.1048 \end{bmatrix}$$

בחרנו את המטריצה U כך שהשונות מסודרת בסדר יורד.

דוגמא לחישוב אחוזי השונות של הנתונים

$$tr(D) = 7614.23 + 427.625 + 98.1048 = 8139.96$$

ניתן לראות כי:

$$tr(S) = 2382.78 + 3106.47 + 2650.71 = 8139.96$$

נקבל:

$$tr(D) = tr(S)$$

עבור הרכיב העיקרי הראשון:

$$\frac{7614.23}{8139.96} = 93.5\%$$

עבור הרכיב העיקרי השני:

$$\frac{427.625}{8139.96} = 5.3\%$$

עבור הרכיב העיקרי השלישי:

$$\frac{98.1048}{8139.96} = 1.2\%$$

SVD ושונות משותפת

נתונה קבוצת הוקטורים $\{\vec{x}_1, \dots, \vec{x}_n\}$ כך ש- $\vec{x}_n \in \mathbb{R}^p$ והממוצע $\sum \vec{x}_k = 0$.

נרצה למצוא בסיס אורתונורמלי עבור תת המרחב הנפרש ע"י וקטורים אלו. במקום, נבחן את המתאם בין הוקטורים הללו. מבחינה גיאומטרית, אנו מחפשים את כיוון השונות המקסימלית.

מבחינה מתמטית אנו מחפשים את הכיוון $\vec{w}$ כך ש:

$$(18) \quad \mu = \max_{\|\vec{w}\|_2=1} \frac{1}{n} \sum_{k=1}^n (\vec{w}^T \vec{x}_k)^2.$$

ניקח את מטריצת השונות המשותפת,

$$S = \frac{1}{n} \sum_{k=1}^n \vec{x}_k \vec{x}_k^T$$

אזי,

$$\vec{w}^T S \vec{w} = \frac{1}{n} \sum_{k=1}^n (\vec{w}^T \vec{x}_k) (\vec{x}_k^T \vec{w}) = \frac{1}{n} \sum_{k=1}^n (\vec{w}^T \vec{x}_k)^2.$$

כלומר,

$$\mu = \max_{\|\vec{w}\|_2=1} (\vec{w}^T S \vec{w}).$$

כעת נראה כי באמצעות פירוק SVD של מטריצה:

$$A = \frac{1}{\sqrt{n}} [\vec{x}_1, \dots, \vec{x}_n]$$

ניתן לחשב בקלות את הכיוון של השונות המקסימלית.

נציג את מטריצה A על ידי פירוק SVD מצומצם:

$$(19) \quad A = \frac{1}{\sqrt{n}} \sum_{j=1}^r \sigma_j \vec{u}_j \vec{v}_j^T$$

אזי,

$$(20) \quad S = \frac{1}{n} \sum_{k=1}^n \vec{x}_k \vec{x}_k^T = \frac{1}{n} A A^T = \frac{1}{n} \left(\sum_{j=1}^r \sigma_j \vec{u}_j \vec{v}_j^T \right) \left(\sum_{i=1}^r \sigma_i \vec{u}_i \vec{v}_i^T \right)^T =$$

$$\frac{1}{n} \sum_{i,j=1}^r \sigma_j \sigma_i \vec{u}_j \vec{v}_j^T \vec{v}_i \vec{u}_i^T = \frac{1}{n} \sum_{j=1}^r \sigma_j^2 \vec{u}_j \vec{u}_j^T$$

מזהות (20) אנו רואים ש- $\vec{u}_j$ זהו וקטור עצמי של מטריצת השונות המשותפת S עם ערך עצמי σ_j^2

עבור $j = 1, \dots, r$ ואפס עבור $j = r+1, \dots, p$.

כלומר המטריצה $U = [\vec{u}_1 \dots \vec{u}_p]$ היא מטריצה מלכסנת של מטריצה S,

$$(21) \quad U^T S U = \frac{1}{n} \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 \\ \vdots & \ddots & & \vdots \\ 0 & \vdots & \sigma_r^2 & 0 \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

ניקח כעת וקטור יחידה $\vec{w}$ ב- $\mathbb{R}^p$ ונחשב $\vec{w}^T S \vec{w}$.

בבסיס האורתונורמלי $\{\vec{u}_1, \dots, \vec{u}_p\}$ נכתוב $\vec{w} = y_1 \vec{u}_1 + \dots + y_p \vec{u}_p$, כאשר $y_j = \vec{w}^T \vec{u}_j$, ומתקיים $\|\vec{w}\|_2^2 = y_1^2 + \dots + y_p^2$.

אזי,

$$\vec{w}^T S \vec{w} = \frac{1}{n} \sum_{j=1}^r \sigma_j^2 (\vec{w}_j^T \vec{u}_j) (\vec{u}_j^T \vec{w}_j) \leq \frac{1}{n} \sum_{j=1}^r \sigma_j^2 y_j^2 \leq \frac{\sigma_1^2}{n} \sum_{j=1}^r y_j^2 \leq \frac{\sigma_1^2}{n} \sum_{j=1}^p y_j^2 = \frac{\sigma_1^2}{n}$$

נבחר את $\vec{w} = \vec{u}_1$ ואז,

$$\vec{u}_1^T S \vec{u}_1 = \frac{1}{n} \sigma_1^2 \vec{u}_1^T \vec{u}_1 = \frac{\sigma_1^2}{n}$$

כלומר, $\max_{\|\vec{w}\|_2=1} (\vec{w}^T S \vec{w})$ מתקבל ב- $\vec{u}_1$ הוקטור הראשון בפירוק ה-SVD (19), או במילים

אחרות $\vec{u}_1$ בכיוון של השונות המקסימלית σ_1^2 .

אם נמשיך באופן זה, ניתן לסכם כי הוקטור העצמי הראשון של מטריצת השונות המשותפת S נותן את כיוון השונות המקסימלית σ_1^2 . הוקטורים העצמיים $\vec{u}_2, \dots, \vec{u}_r$ מצביעים על הכיוונים של השונות בסדר יורד.

שימוש במטריצת השונות המשותפת לחישוב ההסתברות שמדידה נופלת בתוך אזור הפנים הממוצעות

נניח כי קיים מאגר תמונות המכיל מידע אודות הרוחב והגובה של פנים אנושיות, ואנו רוצים לדעת אם פנים מסוימות יכולות להוות חלק מהמאגר.

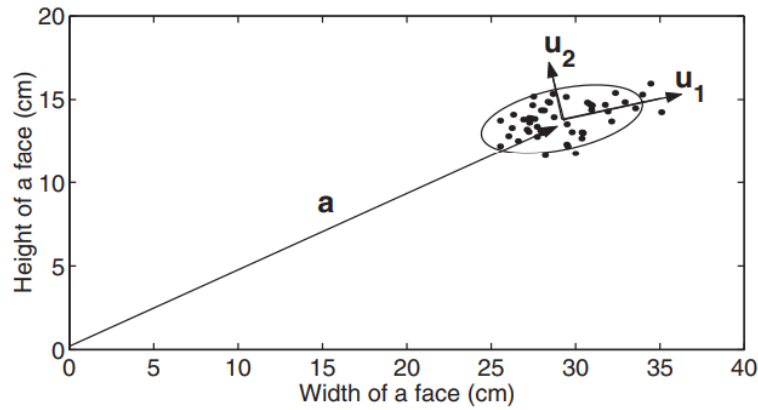
לשם חישוב ההסתברות שמדידה של פנים נופלת בתוך טווח הפנים הממוצעות - האליפסה A, נעשה שימוש בפירוק SVD.

נציג את הבעיה במרחב הדו-ממדי, כאשר $p=2$. ניתן להכליל את הבעיה למרחב n ממדי.

$U_S = [\vec{u}_1 \ \vec{u}_2]$ מטריצה המגדירה לנו את כיווני השונות המקסימלית, והם הצירים העיקריים של אליפסה A, כפי שניתן לראות באיור.

האליפסה ממורכזת בנקודה a שהיא הממוצע של נתוני הפנים במאגר.

באופן אינטואיטיבי ניתן להבין כי ככל שמדידה רחוקה יותר מהאליפסה A הסיכוי שהפנים נמצאות במאגר נמוך יותר.



נניח כי מדובר בהתפלגות נורמלית, נגדיר את ההתפלגות שמדידה $\vec{x}$ נופלת בתחום האליפסה A

כאשר $\vec{x} = [x_1 \ x_2]$:

$$(22) \quad p(x) = \frac{1}{2\pi|S|^{1/2}} \exp\left(-\frac{1}{2}(\vec{x} - a)^T S^{-1}(\vec{x} - a)\right)$$

כאשר S היא מטריצת השונות המשותפת, ו-|S| זאת הדטרמיננטה שלה.

ההסתברות שפנים יהיו בתחום A מוגדרת על ידי:

$$P = \int_A p(\vec{x}) d\vec{x}$$

לפי (16) $S = U_s D_s U_s^T$, כאשר $D_s = \text{diag}(\sigma_1^2, \sigma_2^2)$. את מטריצת השונות המשותפת ניתן

$$\text{לרשום גם בצורה } S = U_s \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix} U_s^T$$

נבצע טרנספורמציה, נציב $\vec{x} = U_s \vec{y}$, כאשר $\vec{y} = [y_1 \ y_2]^T$. זאת טרנספורמציה אורתוגונלית ולכן היעקוביאן $J = \det(U_s) = 1$ מכיוון שמטריצה U_s מטריצה אורתוגונלית.

כעת נקבל:

$$\begin{aligned} \vec{x}^T S^{-1} \vec{x} &= (U_s \vec{y})^T S^{-1} (U_s \vec{y}) = \{S^{-1} = S^T\} = \vec{y}^T (U_s^T S^{-1} U_s) \vec{y} = \vec{y}^T \begin{bmatrix} \frac{1}{\sigma_1^2} & 0 \\ 0 & \frac{1}{\sigma_2^2} \end{bmatrix} \vec{y} \\ &= \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}^T \begin{bmatrix} \frac{1}{\sigma_1^2} & 0 \\ 0 & \frac{1}{\sigma_2^2} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \frac{1}{\sigma_1^2} y_1^2 + \frac{1}{\sigma_2^2} y_2^2. \end{aligned}$$

נציב ונקבל את ההסתברות:

$$(23) \quad P = \int_A p(\vec{x}) d\vec{x} = \frac{1}{2\pi} \frac{1}{|S|^{1/2}} \int_{\tilde{A}} \exp\left(-\frac{1}{2}\left(\frac{y_1^2}{\sigma_1^2} + \frac{y_2^2}{\sigma_2^2}\right)\right) d\vec{y}.$$

נבצע טרנספורמציה שנייה, כאשר $z_j = \frac{y_j}{\sqrt{\sigma_j}}$, נקבל כי $y_1 = z_1 \sqrt{\lambda_1}$, $y_2 = z_2 \sqrt{\lambda_2}$.

בנוסף, $dy = \sqrt{\sigma_1 \sigma_2} dz$, היעקוביאן שנקבל $J = \sqrt{\sigma_1 \sigma_2} = \det(S)$, נקבל,

$$(24) \quad \int_A p(\vec{x}) d\vec{x} = \frac{1}{2\pi} \int_{\tilde{A}} \exp\left(-\frac{1}{2}(z_1^2 + z_2^2)\right) d\vec{z}$$

כאשר $\tilde{A}$ היא התמונה של $\hat{A}$ תחת הטרנספורמציה הלינארית.

לסיכום, הטרנספורמציות שביצענו נתנו לנו נוסחה כללית להסתברות שפנים נופלות בתוך אזור הפנים הממוצעות:

$$(25) \quad P = \frac{1}{2\pi} \int_{\tilde{A}} \exp\left(-\frac{1}{2}(z_1^2 + z_2^2)\right) dz$$

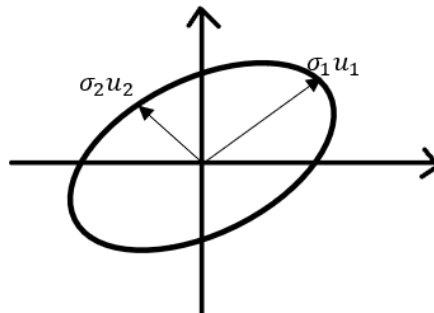
כאשר $\tilde{A}$ היא התמונה של $\hat{A}$ תחת הטרנספורמציה הלינארית.

דוגמא לחישוב הסתברות שמדידה נופלת בתוך אזור הפנים הממוצעות

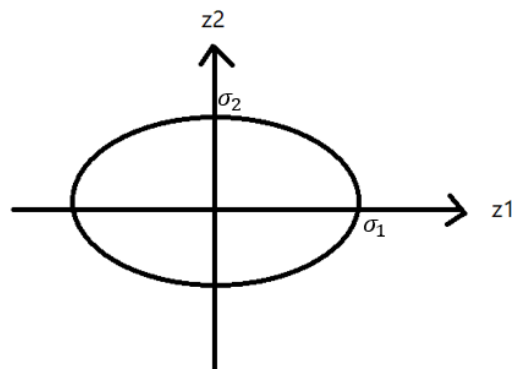
לפי (16),

$$S = U_S \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} U_S^T = \sigma_1 \vec{u}_1 \vec{u}_1^T + \sigma_2 \vec{u}_2 \vec{u}_2^T$$

מתקבלת אליפסה A:



אם נציב $\vec{x} = U_s \vec{y}$ (טרנספורמציה לינארית ראשונה):

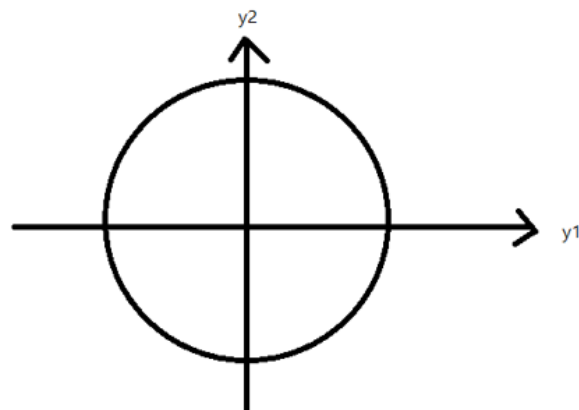


כאשר,

$$\left(\frac{y_1}{\sigma_1}\right)^2 + \left(\frac{y_2}{\sigma_2}\right)^2 \leq 1$$

מטרנספורמציה לינארית שנייה, כאשר:

$$z_2 = \frac{y_2}{\sigma_2}, z_1 = \frac{y_1}{\sigma_1}$$



כאשר רדיוס המעגל שמתקבל הוא 1.

אם ניקח את $\hat{A}$ מעגל ברדיוס 2 אז,

$$P = \frac{1}{2\pi} \int_{\{z_1^2 + z_2^2 \leq 4\}} e^{-\frac{1}{2}(z_1^2 + z_2^2)} dz_1 dz_2$$

נעבור לקואורדינטות פולריות:

$$P = \frac{1}{2\pi} \int_0^{2\pi} \left(\int_0^2 \rho e^{-\frac{\rho^2}{2}} d\rho \right) d\theta = -e^{-\frac{\rho^2}{2}} \Big|_0^2 = 1 - e^{-2} \approx 0.865$$

קיבלנו את ההסתברות P שמדידה נופלת בתוך האליפסה. במילים אחרות, אם המדידה נופלת על או מחוץ לאליפסה המוצגת, הסיכוי שאכן מדובר בפנים הרצויות הוא פחות מ-14%.

ככל שמדידות נופלות רחוק יותר מגבולות האליפסה, הסיכוי שהם מייצגות פנים מהמאגר קטן משמעותית. בדרך זו ניתן יהיה לפסול מספר רב של מתמודדים, אך ההחלטה שצריך לקחת היא את המרחק שמימנו אנו רוצים לפסול. בעצם, מתי אנו עלולים לעבור את הגבול שבו ניתן להחמיץ פנים אפשריות.

ייצוג תמונה במטריצה

השלב הראשון של כל קוד או אלגוריתם לעיבוד תמונה דיגיטלית הוא הפיכת התמונה לייצוג מטריציוני.

בעצם אם קיימת תמונה A כלשהי, היא תהפוך למטריצה הבאה:

$$A = \begin{bmatrix} f(1,1) & f(1,2) & \dots & f(1,n) \\ f(2,1) & f(2,2) & \dots & f(2,n) \\ \vdots & \vdots & \ddots & \vdots \\ f(m,1) & f(m,2) & \dots & f(m,n) \end{bmatrix}$$

כאשר (x, y) הן הקואורדינטות המרחביות של התמונה והפונקציה f מחזירה את עוצמת הפיקסל המסוים הזה.

הערך של הפונקציה מאופיינים על ידי כמות התאורה ומשתקף על ידי האובייקט בתמונה. הערכים m ו- n תלויים ברזולוציה של המצלמה שבה נעשה השימוש, כאשר m ו- n גדלים ככל שהרזולוציה (מספר של מגה פיקסלים) עולה.

ההנחה שהתמונות שצולמו על ידי המצלמה מתקבלות תמיד בפורמט RGB (צבעוני). כאשר צבעי היסוד הם $R=RED$, $G=GREEN$, $B=BLUE$. מודל ה-RGB קשור לאופן שבו אנו תופסים צבע עם G ו- B קולטנים ברשתית שלנו. שילוב של שלושת הערכים נותן לנו $256^3 = 16777216$ צבעים.

תמונות צבעוניות לוקחות יותר זמן לעיבוד, ולכן נהוג לשנות את קנה המידה של התמונה (המטריצה) לטווח גווני אפור. ניתן להשיג זאת על ידי שינוי קנה המידה של ערכי העוצמה כך שייקחו ערכים בין 0 שמייצג את השחור ל-255 במייצג את הלבן.

בתמונה (8 סיביות) בגווי אפור לכל אלמנט בתמונה יש עוצמה שנעשה בין 0 ל-255. תמונה בקנה מידה אפור הוא מה שאנשים מכנים בדרך כלל שחור-לבן. אבל השם מדגיש שתמונה זו תכלול גם הרבה גווניים של אפור.

כדי להפוך תמונה צבעונית לתמונה בגווי אפור ישנן כמה שיטות, נתמקד בשיטה המרכזית והמוצלחת ביותר - שיטת הבהירות. בשיטה זו עלינו לקחת את הממוצע המשוכלל של ערכי ה-RGB עבור כל פיקסל בתמונה. חוקרים הגיעו למסקנה שלכל צבע יש השפעה אחרת על העין ולכן הנוסחה שמתקבלת:

$$gray\ Scale = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$

כאשר אנו רוצים לייצג תמונה בשפת תכנות, כמו למשל Python אנו בעצם נייצג את מטריצת התמונה בווקטור אחד. משרשרים את השורה הראשונה לוקטור, לאחר מכן נשרשר את השורה השנייה לסוף הוקטור, וכך הלאה...:

$$\vec{a} = [f(1,1) \quad f(1,2) \quad \dots \quad f(1,n) \quad f(2,1) \quad \dots \quad f(2,n) \quad \dots \quad f(m,n)]$$

Eigenfaces (פנים עצמיות) - זיהוי פנים ע"י שימוש ב- SVD ו- PCA

הרעיון מאחורי טכניקת פנים עצמיות, הוא לחלץ את המידע הרלוונטי הכלול בתמונות הפנים ולייצג אותו בצורה היעילה ביותר. במקום לעשות השוואה ישירה של פנים, נעשה מניפולציות על מנת לקבל את הייצוגים שלהם.

נניח כי תמונות הפנים מיוצגות ע"י מערכים דו-ממדיים בגודל $m = p \times q$. כאשר m יכול להיות מספר מאוד גדול, אפילו עבור רזולוציה מאוד גסה (כגון $10000 = 100 \times 100$).

ע"י הפיכת המטריצה לוקטור, נהפוך את התמונה שלנו לוקטור באורך m . בדרך זו, m ערכים מתארים לנו תמונה. זאת אומרת שניתן להפוך כל תמונה בגודל $p \times q$ לוקטור במרחב m ממדי. בשיטת פירוק SVD, מצאנו בסיס אורתונורמלי למרחב העמודות של מטריצה.

נניח כי קיימים n וקטורים עם m רכיבים, כאשר כל וקטור בנוי מתמונות פנים ע"י פעולת ערמת העמודות במטריצה שמייצגת את התמונה לוקטור. וקטורים אלו מייצגת את מערך האימון (training set), כל וקטור מסומן ב- $\vec{f}_j$ כאשר $j = 1, \dots, n$. ברור כי לא ניתן לקחת את כל תמונות הפנים של האנושות, אך מערך האימונים מייצגת מגוון רחב של תמונות פנים שמערכת זו עלולה להיתקל בהן. כמובן שנדרוש שכל תמונת פנים שנמצאת במערך שלנו תהיה מיוצגת בצורה ברורה וטובה לנו. לדוגמה, יש צורך להגביל את הסטייה של כל תמונת פנים בודדת מהפנים הממוצעות. ישנם אנשים מסויימים באוכלוסייה שהם נורא ייחודיים, כך שלא יתאימו למערכת זאת ולכן יש לבנות את מערך האימון בצורה זהירה. בדרך כלל, $n \ll m$.

תמונות הפנים צריכות להיות מנורמלות בהתייחס למיקום, גודל, כיוון ועוצמת התאורה. כל תמונות צריכות להיות באותו הגודל, בעלות אותה זווית (ניצב הוא האידאל) ובעלות אותה עוצמת תאורה וכו'. דברים אלו דורשים עיבוד תמונות מקדים.

הפנים הממוצעות מציגות לנו מידע כללי על הפנים, מידע זה חיוני לנו. כמו כן, מכיוון שהערכים בוקטורים המייצגים את תמונות הפנים אינם שליליים ותמונות הפנים מתוקנות מהמקור, ניתן להתייחס לפנים הממוצעות כהטיה אחידה של כל הפנים.

נחסיר את הממוצע מכל התמונות, הממוצע והסטיות המכונים גם הקריקטורות:

$$\vec{a} = \frac{1}{n} \sum_{j=0}^n \vec{f}_j$$

$$\vec{x}_j = \vec{f}_j - \vec{a}$$

כעת נצטרך למצוא בסיס למרחב הנפרש ע"י הוקטורים $\vec{x}_j$. וקטורי בסיס אלו יהפכו לאבני בניין בתהליך שחזור הפנים בהמשך, גם אם הפנים נמצאים במערך האימון וגם אם לא. על מנת לבנות בסיס אורתונורמלי לתת המרחב שנפרש על ידי הפנים במערך האימון, נגדיר מטריצה $X \in \mathbb{R}^{m \times n}$ על ידי,

$$X = \frac{1}{\sqrt{n}} [\vec{x}_1 \quad \dots \quad \vec{x}_n]$$

באמצעות פירוק SVD של X :

$$X = \sum_{j=1}^n \sigma_j \vec{u}_j \vec{v}_j^T$$

ניתן לקרב את X תוך כדי שמירה על עיקרי הנתונים.

כל $1 \leq v \leq n$ ניקח את המטריצה

$$X_v = \sum_{j=1}^v \sigma_j \vec{u}_j \vec{v}_j^T.$$

אז המטריצה

$$X_v X_v^T = \sum_{j=1}^v \sigma_j^2 \vec{u}_j \vec{u}_j^T$$

מקרב את מטריצת השונות המשותפת S . ההפרש בין S לקירוב הוא:

$$S - X_v X_v^T = X X^T - X_v X_v^T = \sum_{j=v+1}^n \sigma_j^2 \vec{u}_j \vec{u}_j^T$$

-1

$$\|S - X_v X_v^T\|_2 = \sigma_{v+1}$$

כלומר אנו מזניחים את הרכיבים העיקריים $\vec{u}_j$ עם השונות קטנות σ_j^2 , ובכך אנו מורידים את ממד המטריצה S .

סדר הקירוב v תלוי בקצב דעיכת השונות $\{\sigma_j^2\}_{j=1}^r$. הוקטורים $\vec{u}_j$ $1 \leq j \leq v$ הם בסיס מממד נמוך ונקראים פנים עצמיות. כמובן שאם מעוניינים לקבל יותר פרטים בתמונות יש להגדיל את המספר v .

תיאור האלגוריתם

1. הפיכת כל מטריצה המייצגת תמונת פנים במערך האימון לוקטור שאורכו מספר הפיקסלים בתמונה, זאת אומרת m שווה למספר הפיקסלים.
2. חישוב פנים ממוצעות, כלומר וקטור הפנים הממוצעות, והחסרתו מכל אחד מהוקטורים שמייצגים תמונת פנים במערך האימון.
3. יצירת מטריצת נתונים X מכל וקטורי הפנים שקיבלנו לאחר החסרת הממוצע.
4. חישוב מטריצת השונות המשותפת ע"י הפעלת SVD על מטריצה X .
5. מציאת וקטורי פנים עצמיות מתוך תהליך ה-SVD.

6. השלכת כל תמונה על כל אחד מוקטורי הפנים העצמיות, לקבלת וקטור מאפיין כשאורכו כאורך וקטור הפנים העצמיות.
7. שחזור התמונה מתוך הפנים העצמיות ובדיקה האם התמונה שהתקבלה דומה לתמונה המקורית שנבדקה.

שימוש בפנים עצמיות

הרעיון מאחורי ייצוג של פנים באמצעות פנים עצמיות הוא בעצם הטלת כל תמונת פנים על המרחב שנפרש על ידי הפנים העצמיות. ליתר דיוק, בהינתן וקטור המייצג פנים $\vec{f}_j$ אנו צריכים להטיל את הוקטור החדש $\vec{f}_j - \vec{a}$ על מרחב העמודות $U_v := [\vec{u}_1 \dots \vec{u}_v]$. זאת אומרת, אנו צריכים לפתור מערכת משוואות ליניארית:

$$U_v \vec{y}_j = \vec{f}_j - \vec{a}$$

בשיטת הריבועיים המזעריים. מכיוון שהעמודות של U_v אורתוגונליות, נקבל כי ייצוג הפנים:

$$\vec{y}_j = U_v^T (\vec{f}_j - \vec{a})$$

ייצוג זה לוכד את כל התכונות שקשורות לפנים $\vec{f}_j$, ובמקום להשוות את תכונות וקטור הפנים $\vec{f}_j$ באופן ישיר, נשווה את התכונות של הוקטור $\vec{y}_j$.

שחזור הפנים העצמיות של וקטור $\vec{f}_j$ ניתן לנו:

$$\tilde{f}_j = U_v \vec{y}_j + \vec{a}$$

עבור מערכת לזיהוי פנים אין צורך בשחזור פנים טוב. אך הכרחי כי התכונות של הייצוג החדש יבדילו בין פרטים שונים. מצד שני, יש לנו עניין בהקטנת הממד של תמונות הפנים, ולכן שחזור ויזואלי טוב מאוד חשוב.

הרעיון הוא לאחסן רק את הייצוג החדש $\vec{y}_j$, ובמידת הצורך לשחזר את הפנים עבורו מתוך מסד הנתונים של תמונות הפנים.

שיטות נוספות לזיהוי פנים

קיימים ארבעה סיווגים לשיטות זיהוי הפנים:

1. שיטות התאמה הוליסטיות.
2. שיטות גיאומטריות.
3. שיטות המרקם המקומי.
4. שיטות למידה עמוקה.

שיטות התאמה הוליסטיות

בגישה זו, האלגוריתמים ההוליסטים מבוססי תת מרחב. אזור הפנים נלקח בחשבון באופן מלא כנתוני קלט לתוך מערכת זיהוי הפנים. אוסף תמונות הפנים מכיל עודף משתנים, שניתן להסרה.

כמו כן, בגישה זו משתמשים באוסף של וקטורי פנים שהם תת המרחב המשמר את סט התמונות המקורי. ע"י וקטורים אלו ניתן לשחזר כל תמונת פנים בתת המרחב. שיטה זו ניתן לחלק לשתי קבוצות: אסטרטגיות לינאריות ולא לינאריות, בהתאם לייצוג תת המרחב.

הטכניקות הלינאריות הנפוצות ביותר לסוג שיטה זאת, הם שיטת הפנים העצמיות (Eigen-faces) בעזרת ניתוח רכיבים עיקריים (PCA) בה עוסקת עבודה זו, ניתוח לינארי של הדיסקרימיננטה (LDA) הידועה בשם "fisherfaces", ניתוח רכיבים עצמיים (ICA) וכו'.

שיטות גיאומטריות

בשיטות אלו, נלקחים בחשבון מאפיינים מקומיים של הפנים כמו עיניים, אף ופה. כאשר תחילה מחלצים את מיקומם של תווי הפנים, נתוני מיקום אלו (מבחינה גיאומטרית) נכנסים לתוך מבנה מסווג. בשיטה זו ניתן להבחין בין שלושה שיטות מיצוי:

- שיטות גטריות המבוססות על קצוות, קווים ועקומות.
- שיטות מבוססות תבניות.
- שיטות התאמה מבניות, הלוקחות בחשבון אילוצי גיאומטריה כאילוץ על תכונות.

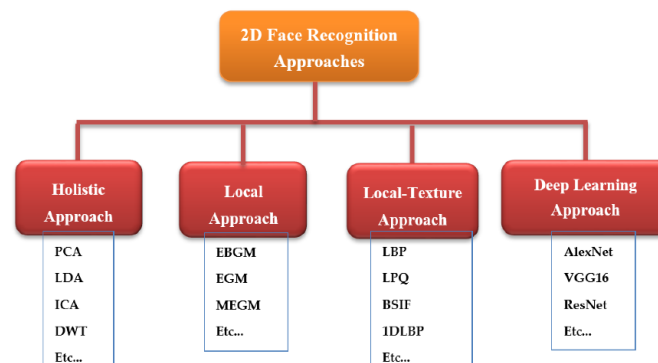
בין הטכניקות הפופולאריות ביותר מסוג זה, התאמת גרפים אלסטיים (EGM), הרחבה של התאמת גרפים אלסטיים (EBGM), התאמת גרפים אלסטיים מורפולוגיים (MEGM) וכו'.

שיטות המרקם המקומי

אסטרטגיות מיצוי תכונה, המתמקדות בידע על המרקם ממלאות תפקיד משמעותי בזיהוי דפוסים ובראייה ממוחשבת. אחד היתרונות הבולטים בשיטה מסוג זה, הוא שהיא עמידה בפני שינויים מונוטוניים בקנה מידה אפור, תאורה גרועה, שונות בבהירות, ואינם זקוקים לסגמנטציה. המטרה של המתאר המקומי היא להפוך את המידע ברמת הפיקסלים לצורה מתאימה, אשר רוכשת את התוכן המשכנע ביותר חסר רגישות להיבטים שונים הנגרמים על ידי וריאציות בסביבה. הטכניקות הנפוצות בשיטה זו, תבניות בינאריות מקומיות (LBP), קוונטיזציה (ייחוס של ערכים מוגדרים למשתנה מסוים) של פאזה מקומית (LPQ) וכו'.

שיטות למידה עמוקה

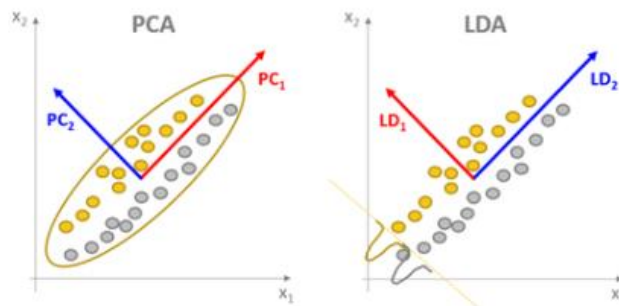
טכנולוגיית למידה עמוקה עיצבה מחדש את נוף המחקר של זיהוי פנים מאז שנת 2014, ע"י פריצות דרך של השיטות "DeepFace" ו-"DeepID". טכניקות מסוג זה ממנפות את הארכיטקטורה ההיררכית ללימוד ייצוג פנים מפלה שיפרו באופן דרמטי את הביצועים העדכניים ביותר וטיפחו מספר רב של יישומים מוצלחים בעולם האמיתי. בשיטות מסוג זה, נעשה לימוד של רמות מרובות של ייצוגי פנים, המתאימים לרמות שונות של הפשטה. הרמות יוצרות היררכיה של מושגים, המראים שונות חזקה לתנוחת הפנים, התאורה וההבעה. עם נתוני אימון נרחבים ויחידות עיבוד גרפיות מודרניות (GPU), שיטות זיהוי פנים עמוקות עלו באופן דרסטי בביצועים וטיפחו יישומים רבים בשנים האחרונות. בין הטכניקות הנפוצות בשיטה זו, "AlexNet", "VGGNet", "ResNet" ועוד.



השוני בין שיטת ניתוח רכיבים עיקריים PCA לשיטת ניתוח לינארי של הדסקרימיננטה LDA

שיטות PCA ו-LDA דומות בכך ששתיהן מפחיתות את ממד המדגם. עם זאת, תחזיות ניתוח רכיבים עיקריים (PCA) אינן מתחשבות בתוויות של המחלקות. ולכן ניתן להתרחק לכיוון אלגוריתם אופטימלי לסיווג. ניתוח לינארי של הדסקרימיננטה (LDA) מוצא השלכה של מחלקות השונות זו מזו. ולכן:

- PCA שומרת על שונות מרבית.
- LDA מאפשרת סיווג מעמדי ע"י מציאת ההשלכה שממקסמת את הפיזור בין המחלקות וממזערת פיזור בתוך הכיתות.



השוני בין שיטת ניתוח רכיבים עיקריים PCA לשיטת ניתוח רכיבים עצמיים ICA

המשותף הוא ששתי הגישות דורשות התאמה אוטומטית של נתוני קלט, כלומר להחסיר כל עמודה בממוצע שלה ולחלק בסטיית התקן שלה. זו אחת הסיבות מדוע PCA הוא בדרך כלל דבר טוב לעשות לפני ביצוע ICA. כפי שידוע לנו PCA היא שיטה לדחיסת המידע, כלומר הפחתת הממד, בעוד ש ICA שואפת להפריד את המידע ע"י הפיכת מרחב הקלט לבסיס עצמאי באופן מקסימלי.



ICA, LDA, PCA

ICA	LDA	PCA
נועד למטרת הפרדה או סיווג מקור עיוור בעזרת רכיבים עצמיים כאשר זהות המחלקה לא נתון.	נועד למטרת סיווג דפוסים במידה וכמות המדגם במערך האימון גדולה.	נועד למטרת הפחתת ממד

מימוש זיהוי פנים ב-python

מימוש השיטה נעשה על מערך אימון שהוא מאגר תמונות פנים של "Yale Face Database". התמונות נאספו בצורה מאוד מעניינת ומגוונת, כולן חתוכות ומיושרות. מאגר התמונות מורכב מתמונות פנים של 38 אנשים, כל תמונת פנים מצולמת ב-9 פוזיציות שונות ו-64 תנאי תאורה שונים. כל תמונה בגובה של 192 פיקסלים ורוחב של 168 פיקסלים (168×192).

כל אחד מתמונות הפנים בספרייה עוצבו מחדש לוקטור עמודה באורך של $192 \cdot 168 = 32256$. במימוש זה נעשה שימוש בתמונות הפנים של 36 האנשים הראשונים במסד הנתונים כמערך האימון בשיטה זו, ובשני הפנים הנותרים נשתמש כתמונות מבחן לזיהוי הפנים.

חלק מספר 1- ייצוג מערך האימון

מכיוון שלקחנו את 36 תמונות הפנים הראשונות למערך האימון נציג אותן במטריצה 6×6 , זאת אומרת תמונה גדולה שתחולק ל-6 שורות ו-6 עמודות, כשבכל תא תהיה תמונת פנים של אדם אחר (דרך נוחה להצגה).

```
import matplotlib.pyplot as plt
import numpy as np
import os
import scipy.io

# In the first part of the code we prepare our database of the
# face images of 36 different people.
numOfRowCol = 6 # The row and column dimension of all
# people's matrix.
plt.rcParams['figure.figsize'] = [10, 10]
plt.rcParams.update({'font.size': 18})

# Database of all the faces.
mat_contents = scipy.io.loadmat(os.path.join(
    'C:\\pythonCode', 'DATA', 'allFaces.mat'))
faces = mat_contents['faces'] # The matrix with the values of
# all faces.
m = int(mat_contents['m']) # Size of columns for each face.
n = int(mat_contents['n']) # Size of rows for each face.
# Database matrix to one dimension vector.
nfaces = np.ndarray.flatten(mat_contents['nfaces'])

allPersons = np.zeros((n*numOfRowCol, m*numOfRowCol))
count = 0
```

#We take the first 6*6=36 people in the database matrix, and insert their image data to "allPersons" matrix.

```
for j in range(numOfRowCol):
    for k in range(numOfRowCol):
        allPersons[j*n: (j+1)*n, k*m: (k+1)*m] =
            np.reshape(faces[:, np.sum(nfaces[:count])], (m,
                n)).T
        count += 1
img = plt.imshow(allPersons) # figure 1.
img.set_cmap('gray')
plt.axis('off')
plt.show()
```

פלט חלק 1



איור מספר 1

חלק מספר 2- הצגת תמונות הפנים עם תנאי תאורה שונים

בחלק זה, נציג את התמונות של כל אדם עם תנאי התאורה השונים מתוך מאגר תמונות הפנים הכללי. שוב, נציג כל אדם במטריצה בגודל 8×8 כגודל 64 תנאי התאורה השונים שצולמו התמונות, כאשר כל איבר הוא תמונת פנים עם תנאי תאורה שונה.

```

personRange = 8          #The row and column dimension of one
person matrix.
for person in range(len(nfaces)):
    subset = faces[:,sum(nfaces[:person]) :
    sum(nfaces[: (person+1)])]
    allFaces = np.zeros((n*personRange,m*personRange))
    count = 0
    # Foreach person in the database matrix we create 8*8=64
    face's images with different light conditions.
    for j in range(personRange):
        for k in range(personRange):
            if count < nfaces[person]:
                allFaces[j*n:(j+1)*n,k*m:(k+1)*m] =
                np.reshape(subset[:,count],(m,n)).T
                count += 1

img = plt.imshow(allFaces)          #figure 2-39.
img.set_cmap('gray')
plt.axis('off')
plt.show()

```

פלט חלק 2



איור מספר 4



איור מספר 3



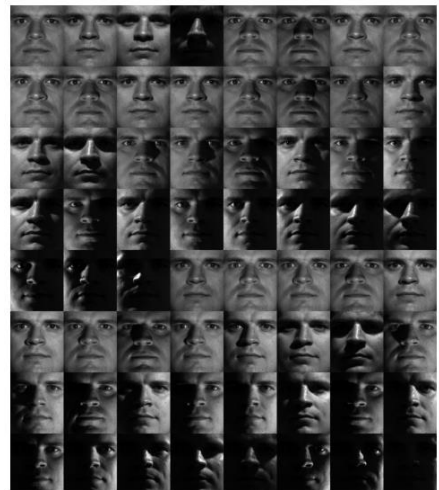
איור מספר 2



איור מספר 7



איור מספר 6



איור מספר 5



איור מספר 10



איור מספר 9



איור מספר 8



איור מספר 13



איור מספר 12



איור מספר 11



איור מספר 16



איור מספר 15



איור מספר 14



איור מספר 19



איור מספר 18



איור מספר 17



איור מספר 22



איור מספר 21



איור מספר 20



איור מספר 25



איור מספר 24



איור מספר 23



איור מספר 28



איור מספר 27



איור מספר 26



איור מספר 31



איור מספר 30



איור מספר 29



איור מספר 34



איור מספר 33



איור מספר 32



איור מספר 37



איור מספר 36



איור מספר 35



איור מספר 39



איור מספר 38

חלק מספר 3 - חישוב הפנים העצמיות (Eigenfaces)

על מנת לחשב את וקטורי הפנים העצמיות נצטרך לחשב תחילה את הפנים הממוצעות.

נחשב את הוקטור הממוצע של כל תמונות הפנים שלנו שמיוצגות בתור וקטורים, ונחסיר את אותו וקטור הממוצע מכל אחד מהוקטורים הללו. לאחר מכן, נחשב את פירוק ה-SVD של הנתונים החדשים לאחר החסרת הממוצע, ובכך בעצם נבצע את תהליך ה-PCA.

נבצע פירוק SVD בגרסה המופחתת (עמ'), לא ניקח את המטריצות המלאות שנקבל, אלא אנו רוצים לקחת רק את החלק שאחראי לערכים הסינגולריים הגדולים.

```
# We use the first 36 people for training data
trainingFaces =
faces[:, :np.sum(nfaces[:numOfRowCol*numOfRowCol])]
avgFace = np.mean(trainingFaces,axis=1) # size n*m by 1

# Compute eigenfaces on mean-subtracted training data
X = trainingFaces -
np.tile(avgFace,(trainingFaces.shape[1],1)).T
U, S, VT = np.linalg.svd(X,full_matrices=0)

#Plot of the average face (avgFace)
fig1 = plt.figure()
ax1 = fig1.add_subplot(121)
img_avg = ax1.imshow(np.reshape(avgFace,(m,n)).T)
img_avg.set_cmap('gray')
plt.title('Average Image')
plt.axis('off')
plt.show()
```

```

#Plot of the first eigenface u1
ax2 = fig1.add_subplot(122)
img_u1 = ax2.imshow(np.reshape(U[:,0],(m,n)).T)
img_u1.set_cmap('gray')
plt.title('Eigenface Image')
plt.axis('off')
plt.show()

```

פלט חלק 3

נראה פלט של הפנים הממוצעות ואת הפנים העצמיות שנקבל.

הפנים הממוצעות שנקבל:



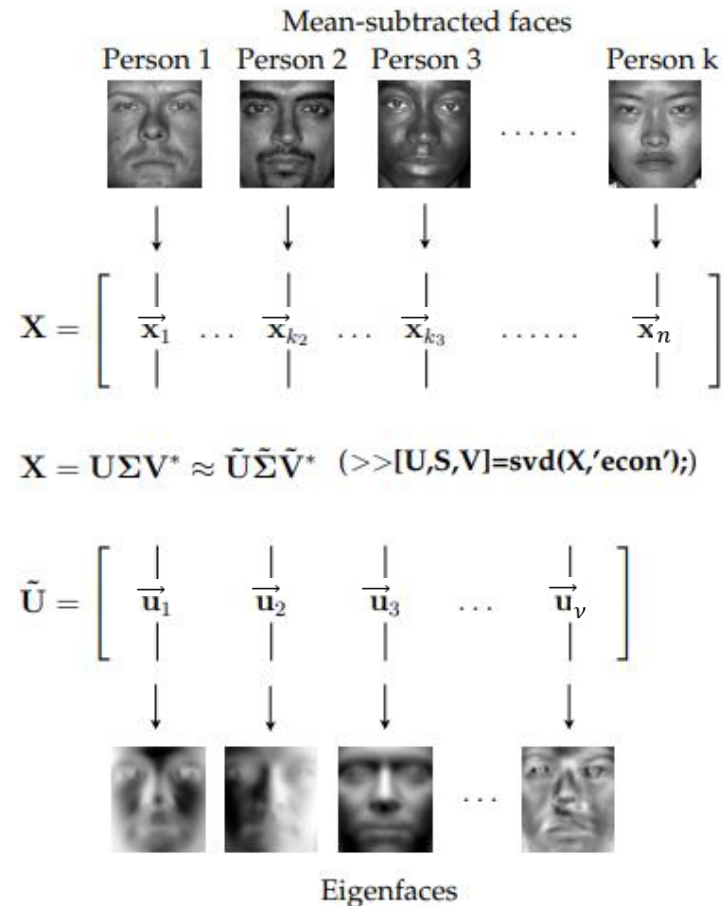
איור מספר 40

אם נחפש ייצוג של הפנים העצמיות $\vec{u}_1$ עבור האדם הראשון שנמצא במאגר שלנו (איור מס' 2), נקבל:



איור מספר 41

אם נפעיל על כל אחד מתמונות הפנים שיש במאגר שלנו (איורים 2-39) נראה כי:



חלק מספר 4 – הטלת תמונות פנים על תת המרחב הנפרש ע"י הפנים העצמיות

בשלב זה, ניקח תמונות פנים שאינה נמצאת במאגר הפנים שלנו. ידוע כי המאגר מידע שלנו מכיל 36 תמונות פנים של אנשים שונים, ולכן ניקח את תמונות הפנים של אדם מספר 37 שקיימת אצלנו ונבצע הטלה שלה על תת המרחב הנפרש ע"י הפנים העצמיות.

```
testFace = faces[:,np.sum(nfaces[:36])] # First face of person
37
plt.imshow(np.reshape(testFace,(m,n)).T)
plt.set_cmap('gray')
plt.title('Original Image')
plt.axis('off')
plt.show()
testFaceMS = testFace - avgFace
r_list = [25, 50, 100, 200, 400, 800, 1600]
```

```
for r in r_list:
    reconFace = avgFace + U[:, :r] @ U[:, :r].T @ testFaceMS
    img = plt.imshow(np.reshape(reconFace, (m, n)).T)
    img.set_cmap('gray')
    plt.title('r = ' + str(r))
    plt.axis('off')
    plt.show()
```

פלט חלק 4

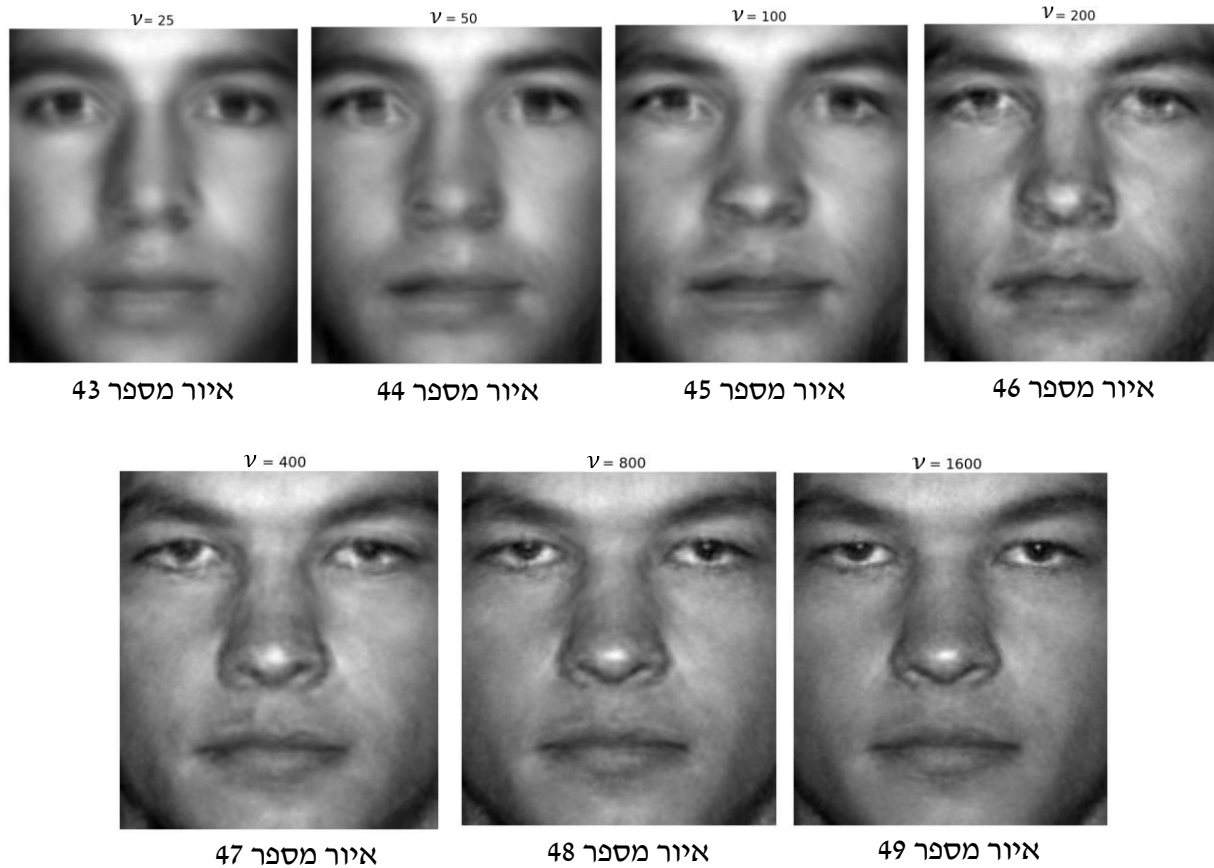
נדפיס את תמונת הפנים המקורית:

Original Image



איור מספר 42

כעת מכיוון שהשתמשנו בפירוק SVD מהצורה המופחתת, ידוע כי קיבלנו מטריצה U מדרגה v כלשהי כאשר $(m \geq v)$, אם ידוע כי m הוא הממד המלא של פירוק SVD. נציג איך תמונות הפנים משתנות כאשר משליכים אותם על תתי מרחב מממדים שונים, זאת אומרת נשנה את v בכל פעם:



ניתן לראות ככל שעולים בממד (v גדל) מקבלים תמונת פנים יותר ברורה ויותר דומה למקור.

בביבליוגרפיה

1. David C. Lay, "Linear Algebra and Its Applications ", 2012, University of Maryland - College Park.
2. Elizabeth A. Compton and Stacey L. Ernstberger, PhD , "Singular Value Decomposition: Applications to Image Processing", 2020, Citations Journal of Undergraduate Research.
3. Gaudenz Boesch, "Deep Face Recognition: An Easy-To-Understand Overview", DEEP LEARNING, viso.ai Web.
4. Insaf Adjabi, Abdeldjalil Ouahabi, Amir Benzaoui and Abdelmalik Taleb-Ahmed, "Past, Present, and Future of Face Recognition: A Review", 23 July 2020, electronics, MDPI.
5. I.T. Jolliffe, "Principal Component Analysis, Second Edition ", 2002, Springer.
6. Iyad Batal, "PCA",
https://www.math.usm.edu/lambers/cos702/cos702_files/docs/PCA.pdf
7. JR Cabansag, Yuxing Chen, Jonathan Griffin, Dunchadhn Lyons, George Preudhomme, "Lecture 13: Face Recognition and LDA", Department of Computer Science Stanford University.
8. Lloyd N. Trefethrn, David Bau 3, "Numerical Linear Algebra", 1997, Siam, Philadelphia.
9. Lourenco Magaia, B. M. Herbst, "Singular Value Decomposition, Eigenfaces, and 3D Reconstructions*", 2004, Society for Industrial and Applied Mathematics, SIAM REVIEW.
10. Sheldon M. Ross, "Introduction to Probability Models, Sixth edition", 1997, Academic Press.
11. Steven L. Brunton, J. Nathan Kutz, "Data – Driven Science and Engineering, Machine Learning, Dynamical System, and Control", 2019, University Printing House, Cambridge CB2 8BS, United Kingdom.
12. Wikipedia, "Covariance Matrix".
13. Xue Yul, "PCA vs ICA vs LDA", 19 May 2012, Slideserve Web.
14. פירוק לערכים סינגולריים, ויקיפדיה.